CONVERGENCE RATE IN ADAPTIVE RADAR

TSC-PD-137-4

John D. Mallett
Irving S. Keed
Frank P. Hopper

February 1976

Technology Service Corporation

Technology Service Corporation

2811 WILSHIRE BOULEVARD • SANTA MONICA, CALIFORNIA 90403 • PH. (213) 829-7411

CONVERGENCE RATE IN ADAPTIVE RADAR

TSC-PD-137-4

John D. Mallett
Irving S. Reed
Frank P. Hopper

February 1976

Fourth Quarterly Report

Submitted to:

THE NAVAL AIR SYSTEMS COMMAND
on
Contract N00019-75-C-0128

TABLE OF CONTENTS

1.0 INTRODUCTION

This is the fourth and final quarterly progress report for a one year study of convergence in adaptive radar. During a series of earlier adaptive radar studies~~[1,2]~~a at TSC, funded by Naval Air Systems Command, a method of optimizing both the angular and doppler response in AMTI (airborne moving target indication radars was described analyzed, and simulated in some detail. These studies showed that adaptive AMTI radar can provide a major improvement in performance in a mixed clutter/ECM environment.

In the first progress report~~[3]~~c the basic adaptive AMTI technique was described, and three adaptive algorithms were compared. These algorithms were sample matrix inversion SMI, inverse matrix update IMU, and the Applebaum loop algorithm. Both SMI and IMU algorithms are shown to be very fast in terms of number of data samples required for convergence. These algorithms are more complicated than the loop technique, though, and are probably best implemented using digital techniques. Section 2 of this report presents results of a computer simulation of analogue to digital (A/D) conversion. A theory predicting the performance loss due to quantization was presented in progress report 3 of this contract and is included as Appendix I. This theory is compared with simulation results in Section 2. The program used to obtain the results of Section 2 is included as Appendix II. In Section 3 of this report some further results are shown for the IMU technique, using the program listed in reference 3.

2.0 SIMULATION OF QUANTIZATION AND LIMITING

In this section we consider the effect of converting the input analogue data to digital data in an adaptive AMTI array, including the ability to reject clutter. A/D conversion introduces two different types of non linearity, one is quantization which produces a step like approximation to a straight line transfer function and the other is limiting which renders the output constant after a particular input level is reached. These effects are important, since the number of bits used in a quantizer is limited for economic reasons and must be as small as is consistent with good performance. The above non-linearities are simulated separately and together using the computer programs in Appendix II. The simulation of quantization and limiting separately allows comparison with the theoretical results for quantization noise given in Appendix I.

The example considered for simulation is an AMTI radar system in which the weights W_{ep} are multiplied by the output of a number E of receiving antenna elements on each of a number P of pulses and then added to obtain the radar output. The non linearities are introduced as shown in Figure 2.1. The weights as in earlier studies can be nearly optimized, using data samples collected by the radar. A single sample of data is defined as the set of returns from a single range bin at each element on each pulse, so that for a 2 pulse MTI with 4 antenna elements, a sample would contain eight complex values. All weights and voltages are complex thereby preserving both phase and

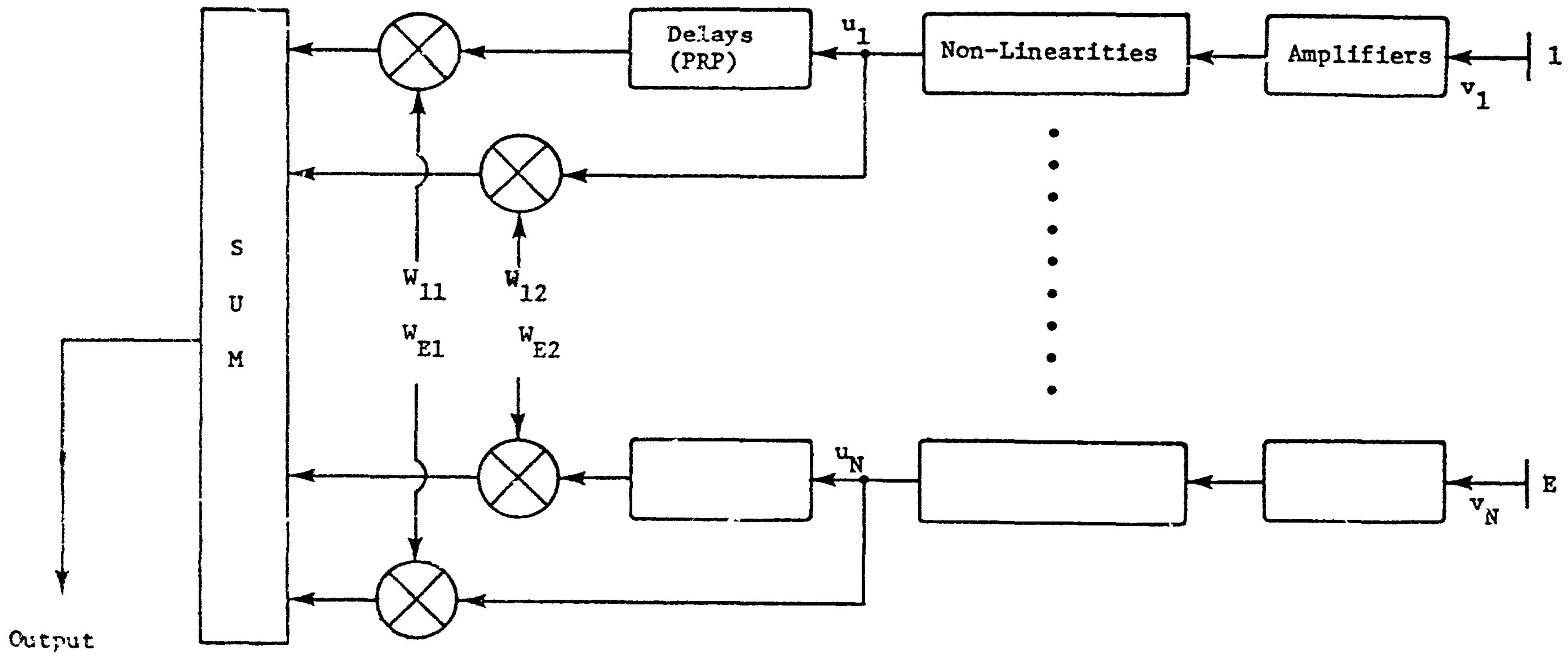


Figure 2.1. Functional Diagram of Adaptive Radar With Non-Linearities.

amplitude information. Each channel shown in Figure 2.1 consists, in fact, of two quadrature channels, one in-phase I, the other 90° out-of-phase Q. On transmission the elements are combined with equal weighting to form a fixed transmit beam.

In a computer simulation where the interference or noise (in this case distributed ground clutter and receiver noise) is known exactly, an ideal covariance matrix M for a linear system is computed as

$$M_{mn} = \sum_{j=1}^{J} V^{*}_{m_j} V_{n_j} \exp\left(\frac{2\pi i}{\lambda} (D_{n_j} - D_{m_j})\right) \tag{2-1}$$

where

V_{m_j} = the voltage at the m^{th} element (pulse) from the j^{th} scatterer

D_{m_j} = distance from scatterer to element

λ = transmitter wavelength

$*$ means complex conjugate

For convenience the indexing over antenna elements and pulses is combined so that V_{ep} becomes V_m where $m = e + (p-1)E$. Included in V is the scatterer amplitude and any other amplitude weighting such as the transmitter antenna pattern and individual element patterns.

The signal-to-clutter or MTI gain is given in vector notation by

$$G = \frac{|WS|^2}{WMW^*} \cdot \frac{1}{(S/C)_o} \tag{2-2}$$

for a set of weights W, where $(S/C)_o$ is the signal to clutter obtained on a single pulse with uniform weighting on receive and transmit. This factor is used for normalization.

As explained in previous references[1-4] the weights that maximize S/C or G are known to be

$$W_o = M^{-1}S^* \qquad (2-3)$$

and

$$G_m = \frac{|W_o S|^2}{W_o M W_o^*} \frac{1}{(S/C)_o} = SM^{-1}S^* \frac{1}{(S/C)_o} \qquad (2-4)$$

where S is the desired signal vector. When non-linearities are present, for example as indicated in Figure 2.1, the clutter covariance matrix cannot always be readily calculated. Instead it can be found by using computer generated data samples to form a sample covariance matrix (or its inverse) as it would be done in an adaptive radar using digital processing. The sample covariance matrix $\hat{M}$ is given by

$$\hat{M}_{mn} = \frac{1}{S} \sum_{j=1}^{S} U_{mj}^* \, U_{nj} \qquad (2-5)$$

where the summation is over independent samples S. The U_{mj}'s are the element voltages after the non-linearities are applied. In the computer simulation, each voltage for a given sample is generated by choosing a different set of Gaussian distributed random numbers for a prescribed set of scatterers. The sample covariance matrix is an approximation to the ideal matrix for a set of scatterers. As it was shown previously[5], the convergence is very fast.

The sample covariance matrix can be used to compute a set of weights. Thus

$$W_s = \hat{M}^{-1} S^* \tag{2-6}$$

The question then is how to test the weights W_s generated from a sample covariance matrix. It is meaningful with a linear system to test the W_s using the ideal covariance matrix since $\hat{M} \rightarrow M$ as the number of samples becomes large. With non-linearities present this is not so.

Two cases are of practical interest. In case 1 a sample matrix is found $\hat{M}_1$ and used to create optimum weights W_1 for $\hat{M}_1$. Then these weights are tested on the same sample data used to obtain the weights. Thus Case 1

$$W_1 = \hat{M}_1^{-1} S^* \tag{2-7}$$

$$G_1 = \left(\frac{W_1 \hat{M}_{1s} W_1^*}{W_1 \hat{M}_1 W_1^*} - 1 \right) \frac{1}{(S/C)_o} \simeq \frac{|W_1 S|^2}{W_1 \hat{M}_1 W_1^*} \; \frac{1}{(S/C)_o} \tag{2-8}$$

where $\hat{M}_{1s}$ and $\hat{M}_1$ are the covariance matrices with and without signal.

Case 2 is similar to the above except that the sample matrix used for testing the weights W_1 is made up of different samples. Thus in Case 2

$$W_1 = \hat{M}_1^{-1} S^* \tag{2-9}$$

$$G_2 \simeq \frac{|W_1 S|^2}{W_1 \hat{M}_2 W_1^*} \frac{1}{(S/C)_o} \tag{2-10}$$

In this formulation the signal is not present in the simulation and is assumed to be unaffected by the non-linearities. Since signal loss is quite small even with hard limiting this is a reasonable approximation.

Case 2 is easier to apply in practice. For Case 1 to be implemented, the data from all elements is stored and used to form a covariance matrix, then the weights derived are used on each range cell individually. This requires more storage than in Case 2, where the weights are used on succeeding range bins as the data arrives.

In practice the weights would be applied to each range bin individually to maximize the signal-to-clutter gain. For simulation purposes one is interested not in individual, but in average performance, therefore one can apply the weights to the already summed covariance matrices $\hat{M}_1$ or $\hat{M}_2$ as in Equation 2.8 or 2.10. This is equivalent to applying the weight to each sample individually and averaging the results.

The relations discussed so far contain the assumption that the signal is not present during the collection of data, i.e. that no signal is present in the covariance matrix used to determine the optimum weights. This situation cannot be strictly true in Case 1, or no signals would be detected. It could be true in Case 2, if the data used for determining the weights were obtained over a region known to contain no signals. From previous simulations[5] it has been found that little degradation in performance occurs when signals are present. If signals are small the dominance of clutter which is present in many range bins compared to the one or two for signals determines the weights as if the signals were not present. If the signal is large detection is not a problem anyway, so that Case 1 as well as Case 2 apply to realizable cases.

Two types of non-linearities are explored in the simulation, quantization which occurs in the conversion of analog data to digital as indicated in Figure 2.2, where a linear function is replaced by a

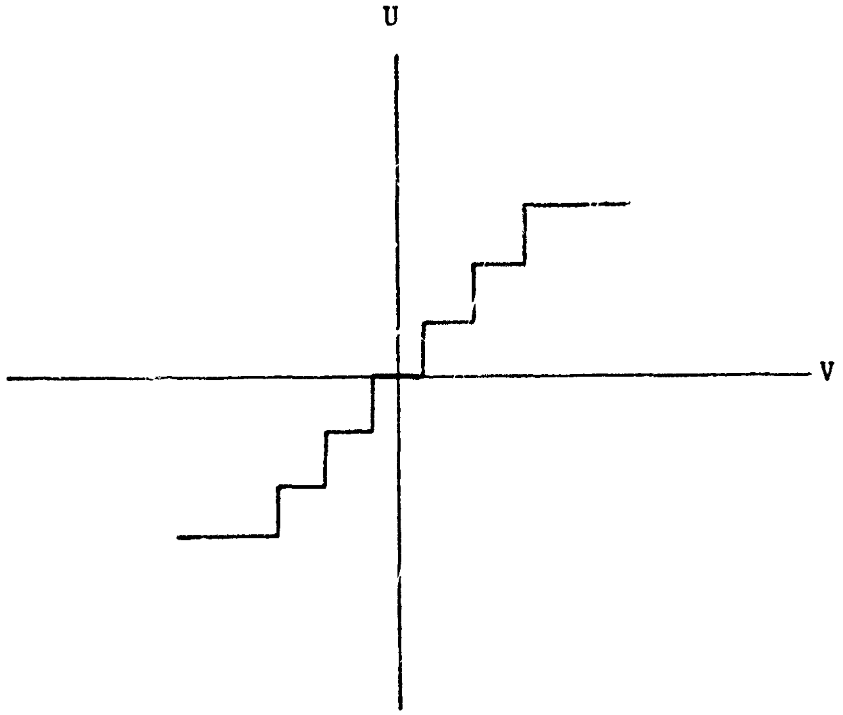


Figure 2.2. A/D Converter With Equal Step Sizes.

stair step, and limiting as also indicated in Figure 2.2. The limiting can be either in the I and Q channels in which case phase distortion occurs, or amplitude (envelope) limiting as might be performed at IF before the quadrature channels are determined. In the latter case the amplitude is limited without causing phase distortion.

2.1 QUANTIZATION

In Table 2.1 are shown results for quantization noise with no limiting for various values of step size Δ divided by RMS input voltage σ. The example used consists of a 4 element antenna with .5λ spacing between elements and 2 pulse MTI. The beam is broadside to the array and is aimed in the direction of travel of the radar which is moving at a rate of .2λ per pulse. Thirty scatterers are distributed uniformly over 180° to simulate ground clutter. Under these conditions the ideal MTI gain as defined earlier is 53.8 dB.

The losses shown are the MTI gain values obtained with quantization divided by the ideal performance and expressed in dB. In this Table 24 or 3N[1] samples are used to form the sample covariance matrix $\hat{M}_1$ which is used to generate the weights W_1. These weights are then tested on $\hat{M}_1$ for Case 1 and on $\hat{M}_2$ which is formed from an equal number but different set of samples. The experiment is repeated 12 times in this Table, i.e. 12 runs and the results of the runs are averaged. The variance as shown is quite small.

Some of the simulation values shown for Case 1 are negative indicating

[1] Where N is the number of degrees of freedom which in this case is 2 x 4 = 8.

Table 2.1

QUANTIZATION LOSS (dB) (NO LIMITING)

2 pulse MTI 4 elements	30 clutter scatterers uniformly distributed over 180°
Platform motion 0.2λ per pulse	Beam angle 0° from velocity vector
Ideal performance (no quantization)	G = 53.8 dB
Samples used in $\hat{M}$ = 24 (3N)	Runs averaged = 12
Theoretical Quant. Loss = L_Q, L_Q'	Theoretical Sample Loss L_s = 1.43 dB

Δ/σ	Case	L_Q	L_Q'	L_Q+L_S	Simulation Loss	Variance
0	1				-1.65	.034
	2	0	0	1.4	1.24	.021
.01	1				.02	.073
	2	1.90	3.47	3.33	2.97	.042
.02	1				3.44	.056
	2	5.05		6.48	6.586	.038
.04	1				7.873	.075
	2	9.91		11.33	10.94	.126
.08	1				13.65	.059
	2	15.54		16.98	16.63	.060

Case 1-W_1 tested on $\hat{M}_1$ Case 2 W_1 tested on $\hat{M}_2$. All values are in dB (-) indicates a gain. Δ/σ = step size/RMS voltage.

an improvement over the so called ideal linear case. This is possible when the weights generated from a particular set of samples are used on the same set as in Case 1. The same weights give poorer results when used on other sets of data generated by the same process.

Two theoretical results have been developed for loss from ideal performance. One loss is due to finite sample size and is given by[5]

$$L_s = 10.\ \log_{10}\left(\frac{S+2-N}{S+1}\right) \qquad (2\text{-}11)$$

where S is the number of samples used, and N is the degrees of freedom, i.e. E · P in this case.

The other loss due to quantization was derived in Progress Report #3 and is reproduced in Appendix 1. This is given by

$$L_Q = 10.\ \mathrm{Log}_{10}\ \frac{S^* M^{-1} S}{S^* M_Q^{-1} S} \qquad (2\text{-}12)$$

where

$$M_Q = M + \frac{\Delta^2}{6} I \qquad (2\text{-}13)$$

Using another approximation, a second formula for the loss was obtained as

$$L_Q' = 10.\ \mathrm{Log}_{10}\left(1 - \frac{\Delta^2}{6}\ \frac{W^* W}{S^* M^{-1} S}\right) \qquad (2\text{-}14)$$

In Table 2.1 the simulation and theoretical losses are shown. It is seen that there is good agreement between Case 2 simulation results and the total losses based on sample size and quantization loss. The approximate results for L_Q' loss do not give such good agreement.

Table 2.2 shows the losses for various numbers of samples in the covariance matrices. Theoretical and simulation results are in very good agreement and the variance seems reasonable for the values chosen.

2.2 LIMITING

In this section limiting without quantization is explored by simulation. Two kinds of limiting are considered. Amplitude or envelope in which the vector amplitude is limited but phase is not distorted, and limiting in the IQ channels where both phase and amplitude are affected. These two types are illustrated in Figure 2.3.

While there are theoretical results for hard limiting, it is difficult to obtain theoretical results for linear operation with the various degrees of soft limiting used in this case. Simulation results are shown in Table 2.3 for limiting in the IQ channels and in envelope, as a function of limit levels divided by RMS input voltage σ.

It is interesting to note that for strong limiting (small values of S/σ), IQ limiting seems to produce the greater loss. This perhaps is due to the greater phase distortion. At values of S/σ producing less limiting, losses seem to be higher when amplitude limiting is used. Also, the variance is quite high for the lower limiting values. Both of these results may be due to the fact that for larger values of S/σ there is less limiting in IQ, thereby causing less chance of a degraded

Table 2.2

QUANTIZATION LOSS (dB) (NO LIMITING)

2 pulse MTI 4 elements	30 clutter scatterers uniformly distributed over 180°
Platform motion 0.2λ per pulse	Beam angle 0° from velocity vector
Ideal performance (no quantization)	S/C - 53.8 dB
Samples used in $\hat{M}$ = 24 (3N)	Runs averaged = 12
Theoretical Quant. Loss = L_Q, L_Q'	Theoretical Sample Loss L_s

				Sample=64(8N) Runs=4 L_S=.5 dB			Sample=32(4N) Runs=9 L_S=1.03 dB			Sample=16(2N) Runs=19 L_S=2.3 dB		
Δ/σ	Case	L_Q	L_Q'	L_Q+L_S	Sim. Loss	Var.	L_Q+L_S	Sim. Loss	Var.	L_Q+L_S	Sim. Loss	Var.
.001	1			-.61		.007		-1.275	.015		-3.02	.068
	2	.024	.024	.52	.33	.001	1.06	.923	.026	2.33	2.35	.073
.01	1				1.22	.03		.55	.025		-1.1	.121
	2	1.91		2.4	2.3	.015	2.94	2.7	.035	4.2	4.01	.094
-1	1				16.9	.007		16.13	.047		14.35	.273
	2	17.42		17.9	18.02	.012	18.46	18.4	.028	19.72	19.37	.102

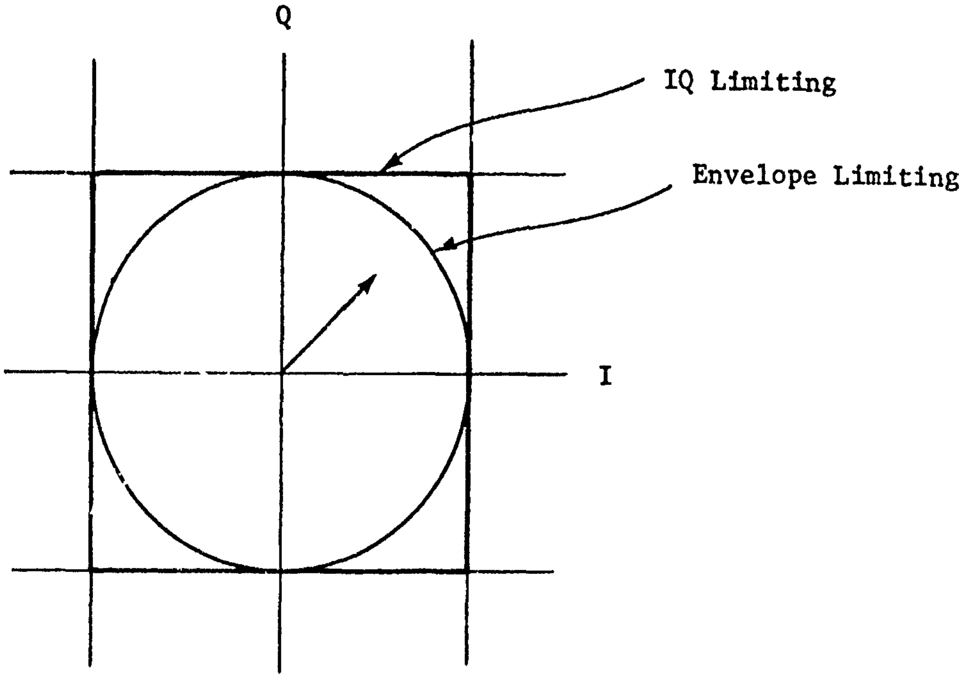


Figure 2.3. Showing IQ and Complex Envelope Limiting.

result. When limiting does occur considerable degradation results. This is usually indicated by drastically increased variance.

The percent limiting, shown in Table 2.3, is obtained by counting the number of times that limiting occurs during simulation. The reason that there is more limiting in the envelope case (the circle in Figure 2.3) is probably that the sample vectors have more "non-limit" area in the complex plan when IQ limiting is used (the square in Figure 2.3).

2.3 LIMITING AND QUANTIZATION

Simulation MTI gains are shown in Figure 2.4 and 2.5 for quantization and limiting combined as a function of normalized saturation level S/σ (σ = RMS input voltage). The results are shown for 8, 9 and 10 bits. As S/σ increases, limiting decreases but the quantization step size Δ increases so that there are optimum values of S/σ. Since quantization noise degrades performance slowly compared to limiting, there is a fairly broad region of S/σ values over which to operate.

There is little difference as indicated by Figure 2.4 and 2.5 between IQ and envelope limiting in the region of small limiting as might be expected. With strong limiting envelope limiting shows better but highly degraded performance.

In Figures 2.6 and 2.7 results are shown for a case with higher MTI gain. In this case the scan angle is 90° to the flight path, and the interpulse motion is close to $\lambda/4$, the velocity and angle for which perfect platform motion compensation is possible. MTI gain is plotted in this section rather than losses to illustrate performance at different gain levels. The Case 1 performance in which W_1 is used with $\hat{M}_1$ rather than $\hat{M}_2$ (Case 2), exceeds the steady state or ideal performance. This

Table 2.3

Simulation Results for Limiting Only

2 pulse MTI   4 elements   30 scatters

Platform motion = 0.2λ/pulse   Beam angle 0.0

Ideal G = 53.8 dB

Sample = 64 (8N)   Runs = 4

		I and Q			Envelope		
S/σ	Case	Loss(dB)	Variance	% Limiting	Loss(db)	Variance	% Limiting
1.5	1	30.5	.015	27	21.3	.039	32
	2	31.7	.001		22.4	.014	
2.0	1	24.0	.07	9	18.4	.056	14
	2	26.7	.17		20.6	.015	
2.5	1	12.3	.533	2.66	12.2	.162	4.33
	2	15.9	1.25		16.1	.023	
3.0	1	2.34	.646	.39	2.91	.82	.86
	2	4.63	.859		5.74	.92	
3.5	1	.632	.01	.04	.009	.157	.23
	2	.28	.002		1.4	.128	

S/σ = Saturation level/RMS voltage.

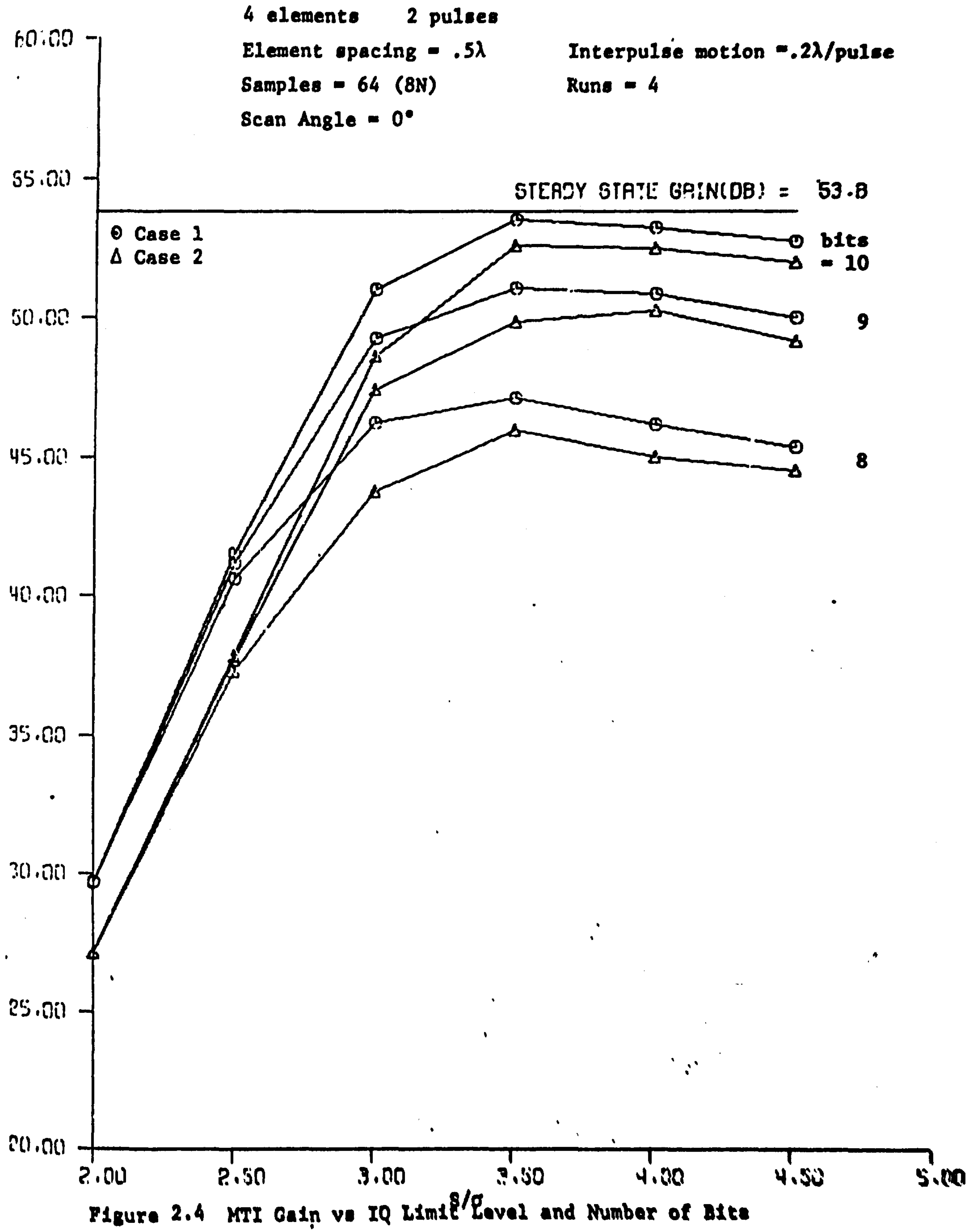


Figure 2.4 MTI Gain vs IQ Limit Level and Number of Bits

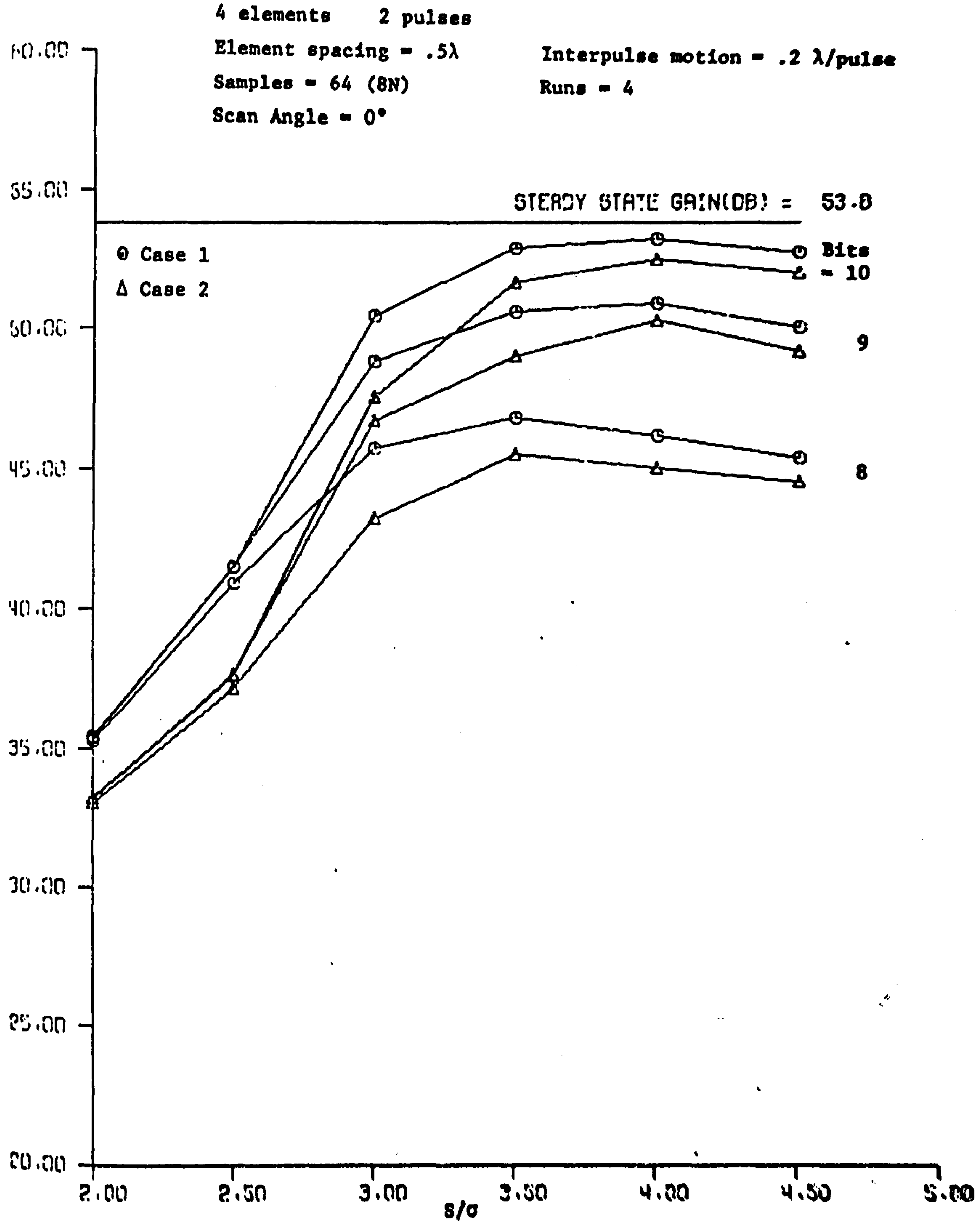


Figure 2.5 MTI Gain vs. Envelope Limit Level and Number of Bits

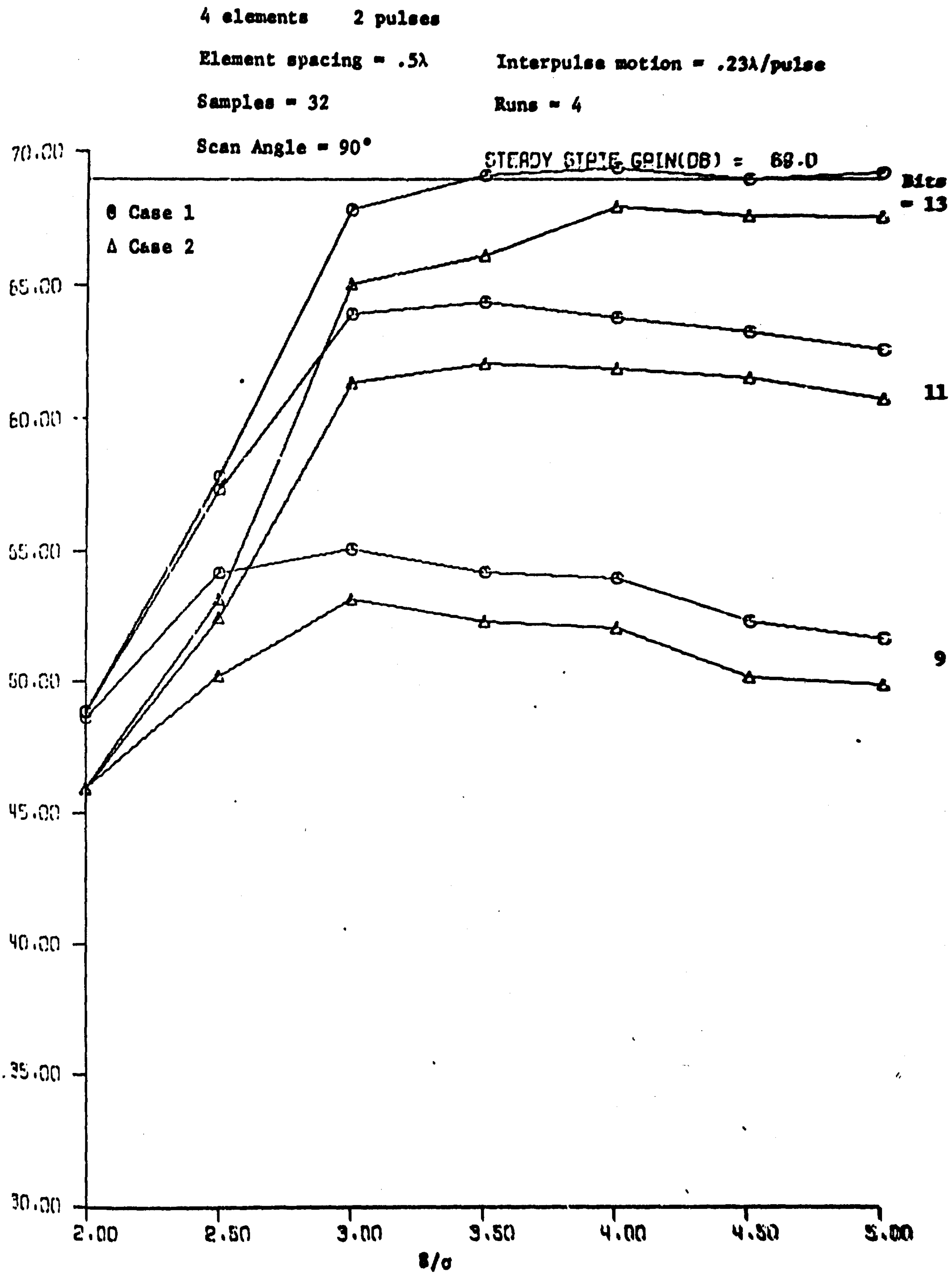


Figure 2.6 MTI Gain vs. IQ Limit Level and Number of Bits

4 elements   2 pulses

Element spacing = .5λ   Interpulse motion = .23λ/pulse

Samples = 32   Runs = 4

Scan Angle = 90°

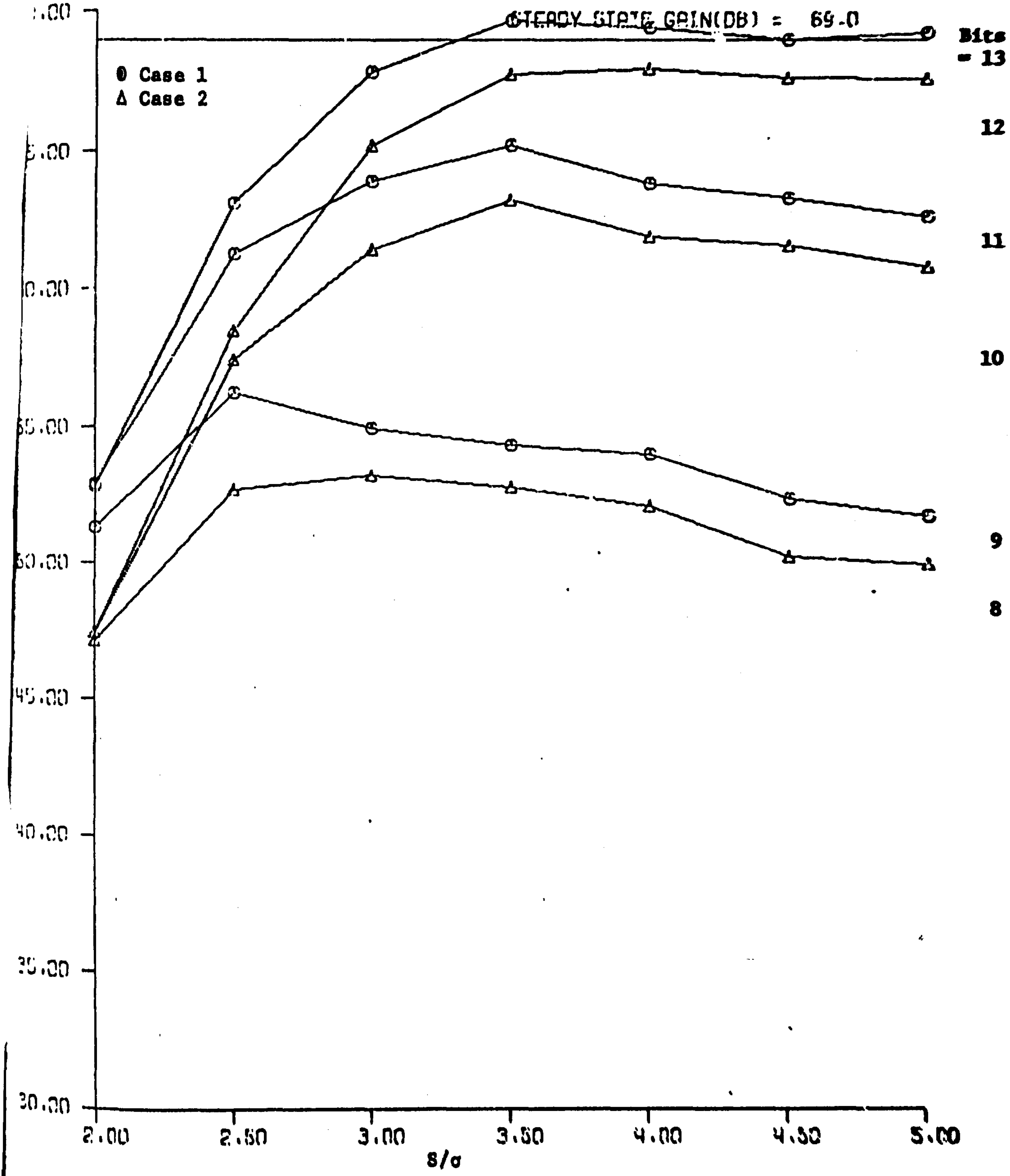


Figure 2.7   MTI Gain vs. Envelope Limit Level and Number of Bits

is possible since the weights are determined for a particular set of samples and used on the same sample set.

In Figure 2.8 the same example is used as in Figures 2.6 and 2.7 but the range of S/σ is extended and different numbers of bits are used. This Figure shows clearly, that while there is an optimum ratio of limit level S to average voltage σ, the optimum is quite broad, and performance declines slowly with increasing values of S/σ. This shows that it is clearly better to operate in a region of low limiting, and to bias the operating point toward the high side of S/σ when σ is changing rapidly or is not accurately known.

Figure 2.9 shows similar results for a lower performance level.

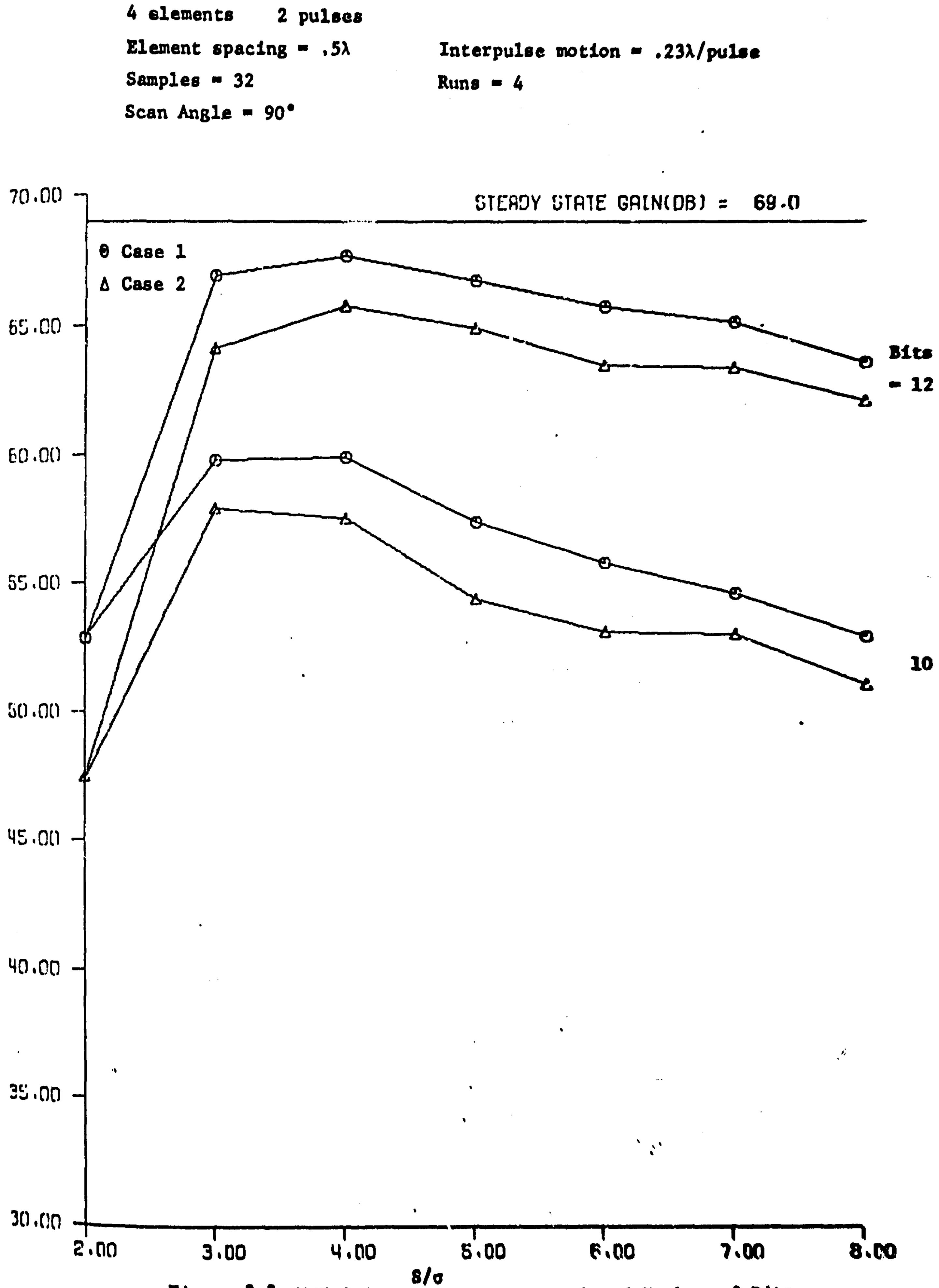


Figure 2.8 NTI Gain vs. IQ Limit Level and Number of Bits

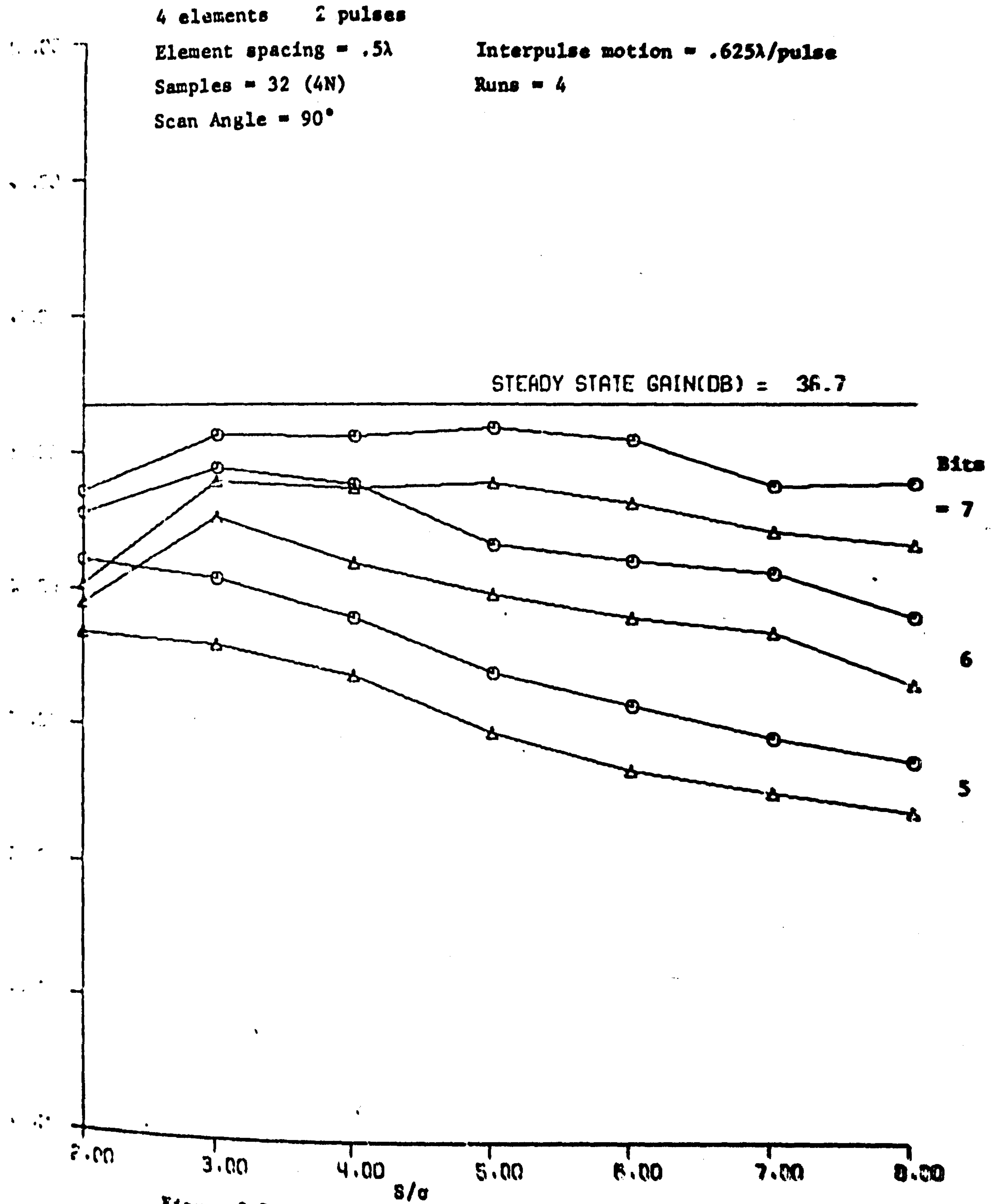


Figure 2.9 MTI Gain vs. IQ Limit and Number of Bits

3.0 SOME CONVERGENCE EXPERIMENTS

In the first progress report[3] the convergence rates of three algorithms were compared using a computer simulation. The sample matrix inversion (SMI) and inverse matrix update (IMU) techniques were shown to be comparable and very fast when measured by the number of independent data samples required to converge. The Applebaum loop, while much simpler to implement, converges much more slowly requiring hundreds of samples, in the example shown, compared to two to four times the number of degrees of freedom, N, required by SMI and IMU.

The SMI technique is known to require the least number of samples under all conditions, and requires only N(N+1)/2 complex multiplications to form $\hat{M}$[5]. These properties are invariant over all problems, thereby making the technique extremely powerful.

To invert the matrix $\hat{M}$ for SMI requires about $N^3/3$ multiplications or divisions. There are several ways of doing this, i.e. solving the set of simultaneous equations required to obtain the weights, or inverting the matrix by various methods. Computational problems may be involved in the SMI or IMU technique when high speed and limited numbers of bits is a requirement.

The IMU algorithm is simple to program, requiring only about $1.75\ N^2 + 2.75\ N$ complex multiplications, per data sample. This can be seen from the recursive equation

$$M_{j+1}^{-1} = \frac{1}{(1-\alpha)} M_j^{-1} \frac{-\alpha}{1-\alpha} \frac{(M_j^{-1}V^*)(V_T M_j^{-1})}{(1-\alpha)+\alpha(V_T M_j^{-1} V^*)} \qquad (3\text{-}1)$$

where the sample matrix is given by

$$M_{j+1} = (1-\alpha)M_j + \alpha V_j^* V_{jT} \qquad (3\text{-}2)$$

Basically $M_j^{-1}V^*$ is formed using N^2 multiplications, $V_T M_j^{-1}$ is formed by taking its conjugate, and to obtain $V_T M_j^{-1} V^*$ requires a further N multiplications. Since M_j^{-1} is Hermitian, multiplying $M_j^{-1}V^*$ by $V_T M_j^{-1}$ requires N(N+1)/2 multiplications. Finally multiplying by the constants provides for the reamining multiplications.

The simplicity of implementing IMU makes it interesting to see how many samples are required for convergence both as a function of α and of the starting values of M_j^{-1}. For fixed values of α an exponential weighting is applied to the incoming data samples as they are used to update the matrix. When $\alpha = \frac{1}{j+1}$, equal weighting is applied to the data samples used to update M^{-1}.

The starting values of M^{-1} are less easily parameterized since they could be anything from an arbitrary choice to a choice which is close to the correct value. To explore IMU as a function of the initial matrix M_o^{-1} one wishes to construct a problem related to a practical radar situation in which external interference (clutter, jamming etc.) is changing. Also one would like the weights to change in such a manner that a required performance is maintained. For example, one might require the weights to adjust as the antenna rotates between pulse groups. This was treated in the first progress report[3]. For another example, one might require the weights to change with the changes in clutter with range as treated in a previous TSC study[6] The latter might be caused by terrain changes such as a land to sea interface.

In these experiments antenna step scanning was assumed in such a manner that the weights are optimized at one angle, then the antenna is rotated to a new angle and held fixed while new samples are taken. In Figure 3.1 the IMU process is started with the diagonal having the same values as the diagonal of the inverted matrix IM_o^{-1} for the previous angle. The inverse matrix is then updated with the new data, using $\alpha = \frac{1}{j+1}$, i.e., giving equal weight to each new sample.

A large rotation angle 5° was chosen to accentuate the missmatch. The results for the IMU and SMI techniques are seen to be quite similar. The loop convergence performance is also shown for the case where steady state performance was reached at the previous angle. The gain/TAU ratio is adjusted so that control loop noise in the steady state increases the total output noise by 25%[7].

It seemed reasonable that if the entire inverse matrix for a previous starting angle were used, that the results would be equally good if not better. Figure 3.2 shows that this is not the case, at least for this example. The IMU performance was considerably degraded when the ideal M_o^{-1} for the previous angle was used as the initial matrix.

Figures 3.3 and 3.4 show that using the previous inverse matrix diagonal even when it is multiplied by .1 or 10 gives extremely good agreement between the SMI and IMU techniques.

Figure 3.1 ADAPTIVE ARRAY/DOPPLER PROC.

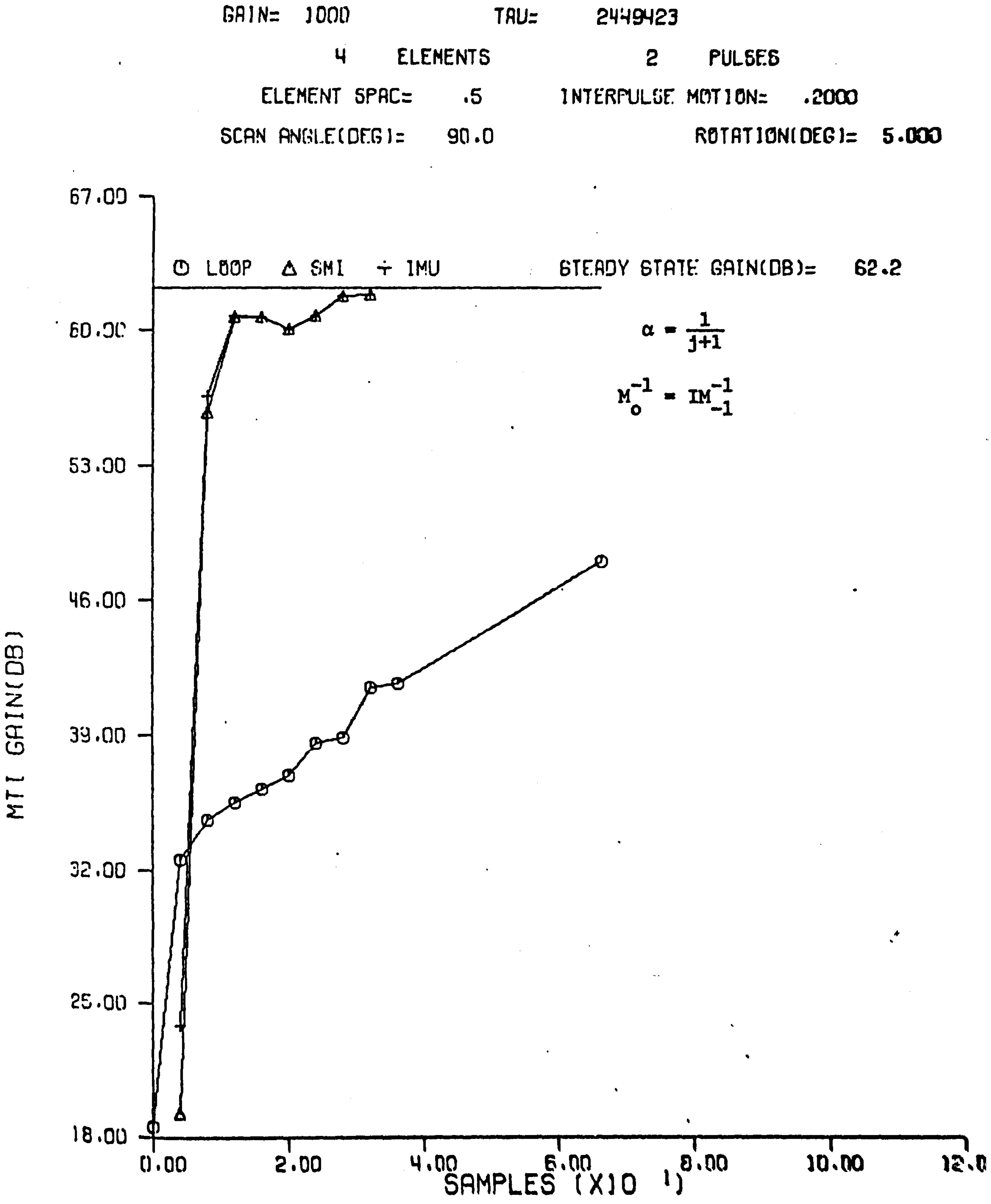


Figure 3.2 ADAPTIVE ARRAY/DOPPLER PROC.

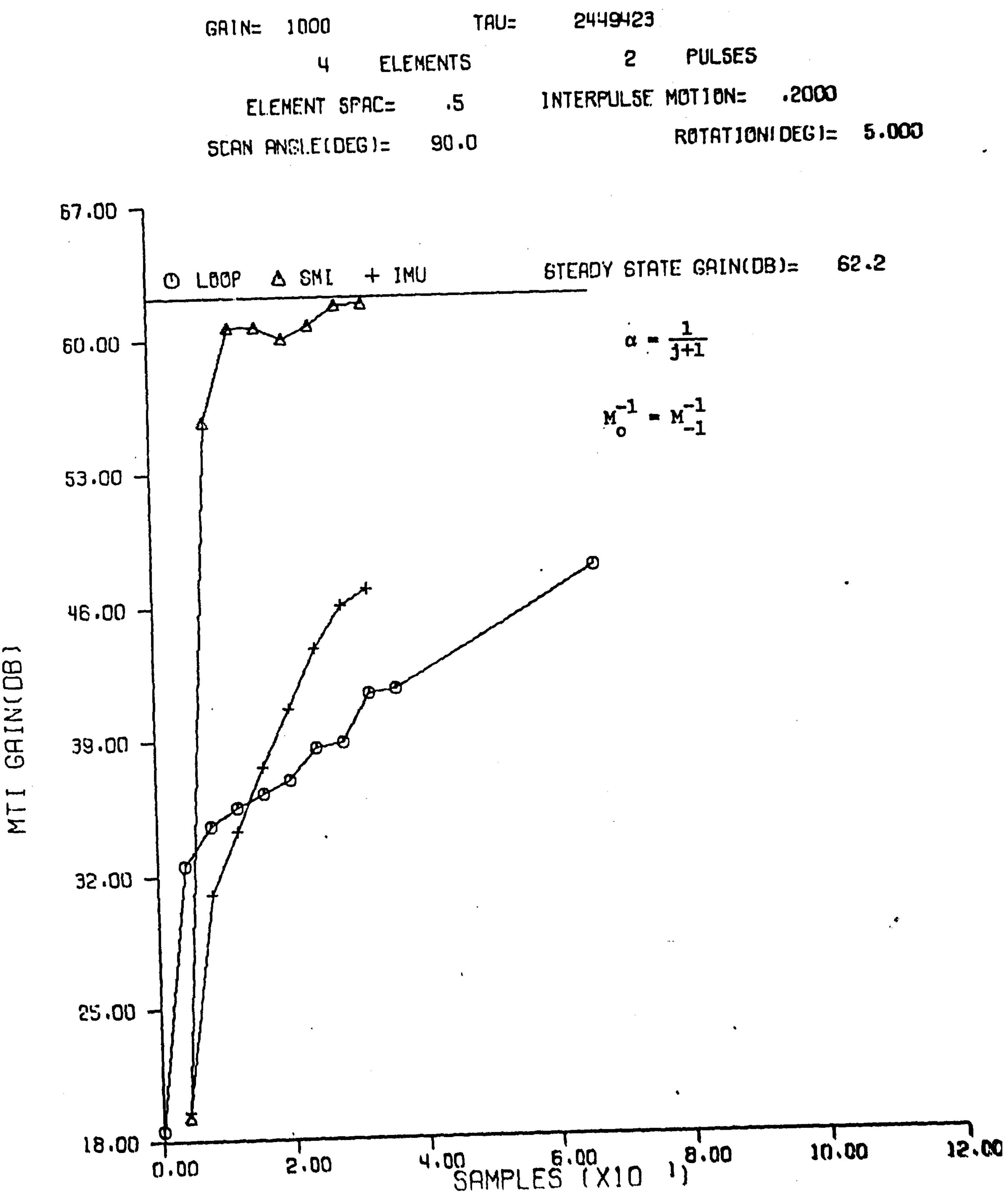


Figure 3.3 ADAPTIVE ARRAY/DOPPLER PROC.

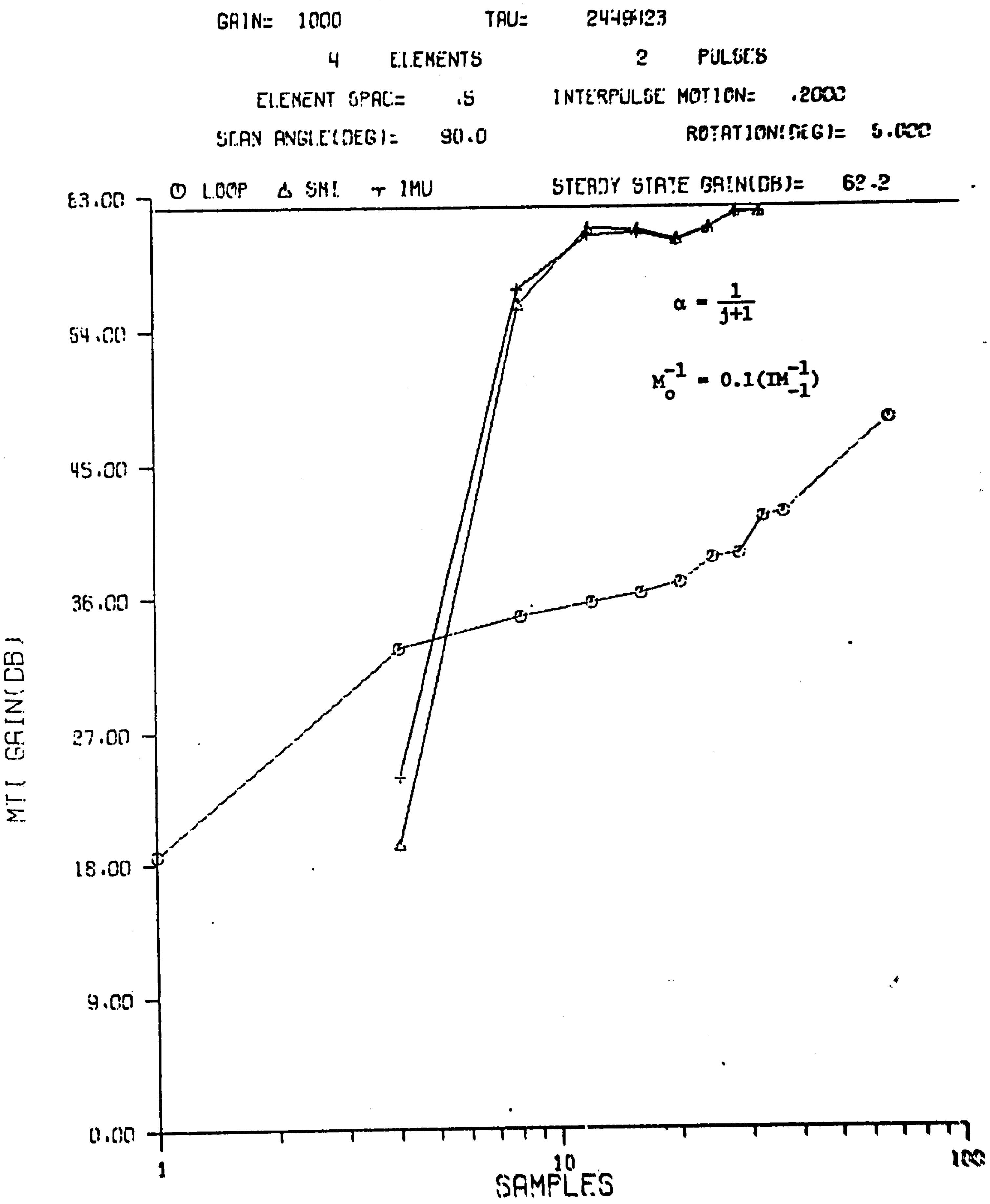
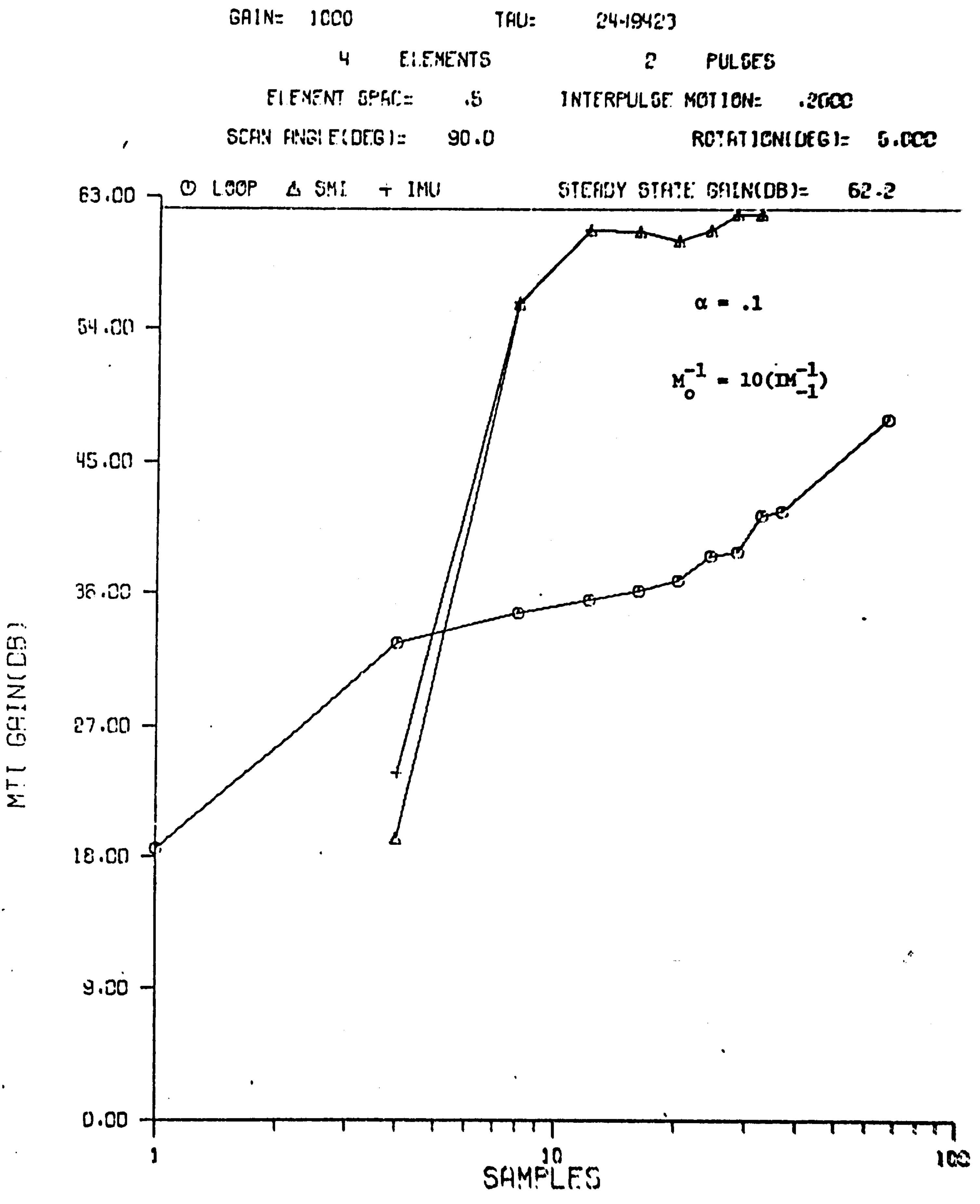


Figure 3.4 ADAPTIVE ARRAY/DOPPLER PROC.



In Figure 3.5 - 3.7 different values of α are shown using the diagonal of the previous updated matrix.

These figures indicate that constant values of α are much less desirable than the varying value which gives equal weighting to the samples. This is what would be expected in the step scan example, assumed in this case. If a steady scan rate were used and data were taken from many range traces, then an optimum constant value for α could be found.

The simulation program ADAPTM6 listed in the first progress report[3] was used in these experiments.

Figure 3.5 ADAPTIVE ARRAY/DOPPLER PROC.

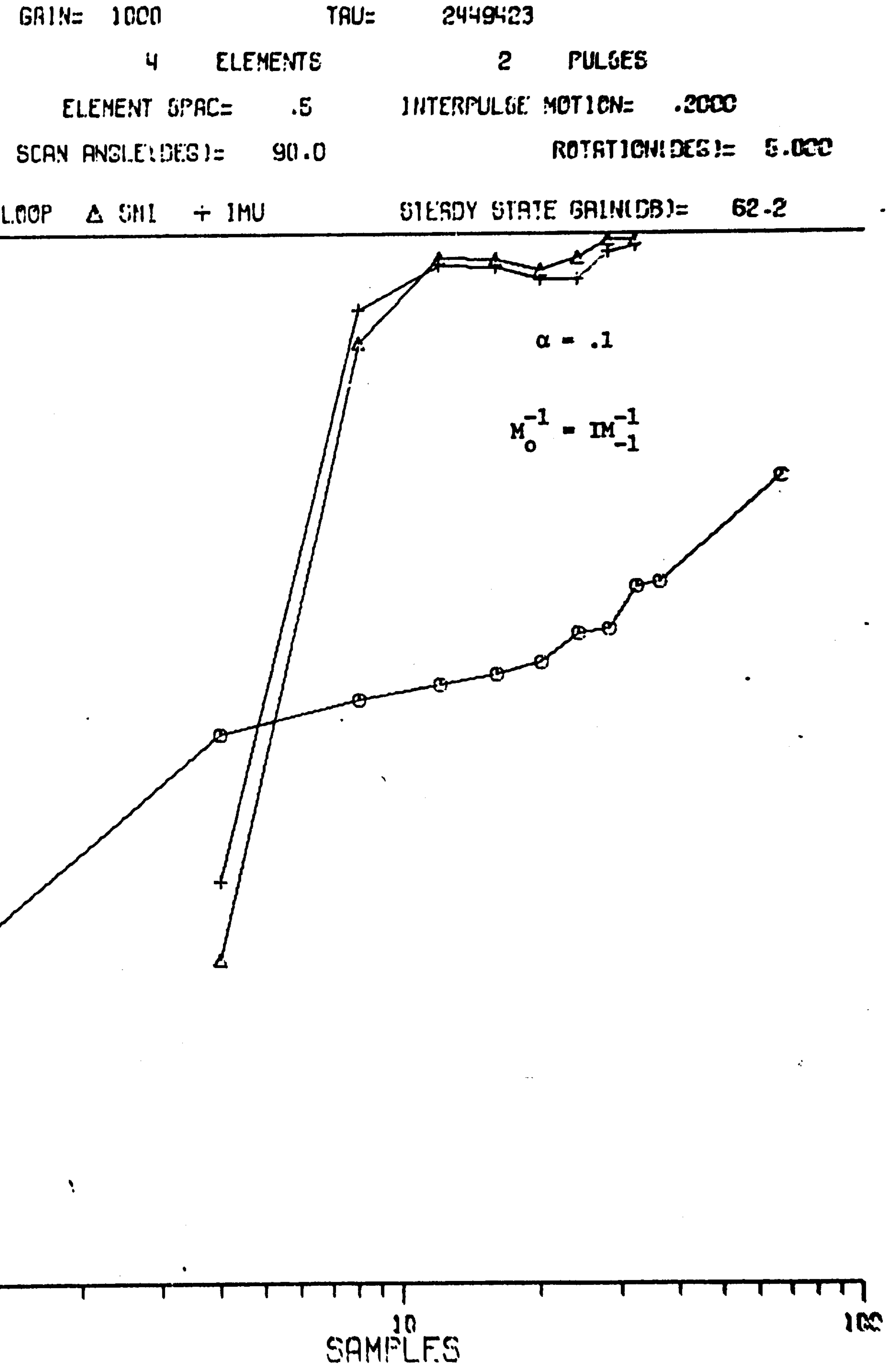
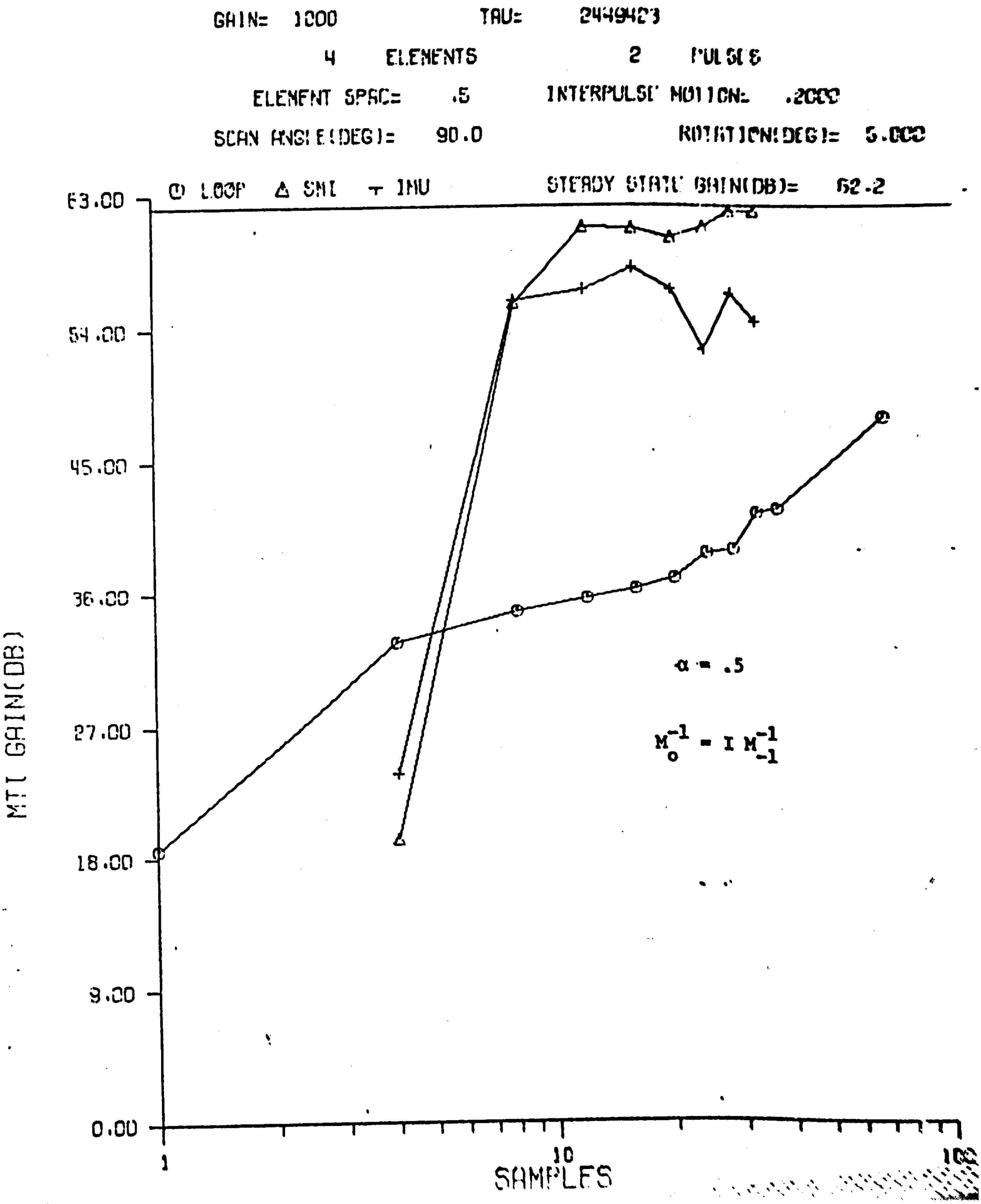


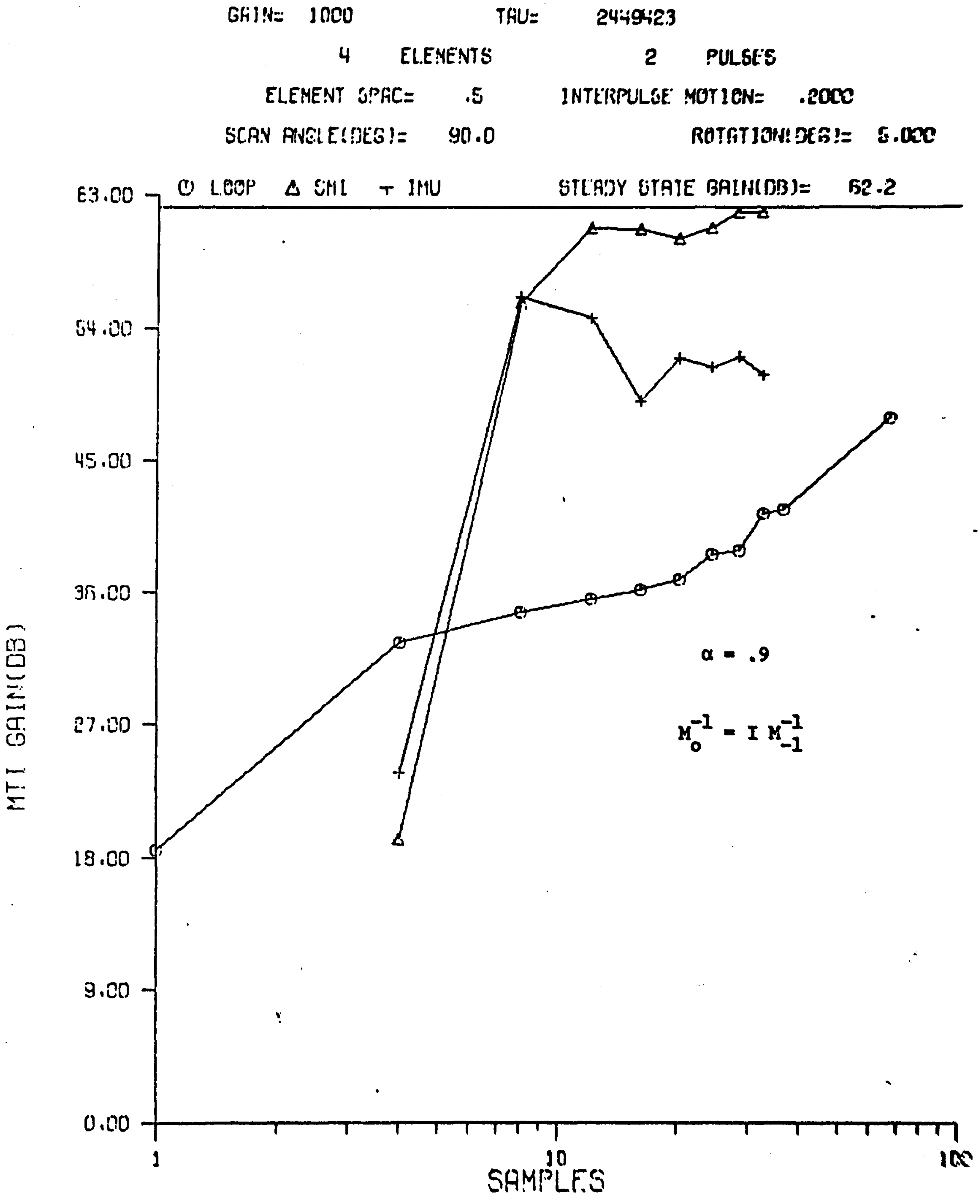
Figure 3.6 ADAPTIVE ARRAY/DOPPLER PROC.



$\alpha = .5$

$M_o^{-1} = I M_{-1}^{-1}$

Figure 3.7 ADAPTIVE ARRAY/DOPPLER PROC.



REFERENCES

1. "Adaptive Array Techniques for Coherent Airborne Radars," Technology Service Corp., TSC-PD-030-1, 26 February 1970, Final Report on Contract N00019-69-C-0662.

2. "ADAPTAR Space-Time Processing in Airborne Radars," Technology Service Corp., TSC-PD-061-2, 24 February 1971, Final Report on Contract N00019-70-C-0387.

3. "Convergence Rate in Adaptive Radar", TSC-PD-137-1, 27 February 1975.

4. L. E. Brennan and I. S. Reed, "Theory of Adaptive Radar", IEEE Trans. of Aerospace & Electronic Systems, VOL. AES-9, No. 2, March 1973.

5. I. S. Reed, J. D. Mallett, L. E. Brennan, "Rapid Convergence Rate in Adaptive Arrays," IEEE Trans. of Aerospace & Electronic Systems, Vol. AES-10, No. 6, Nov. 1974.

6 I. S. Reed, J. D. Mallett, L. E. Brennan, "Adaptive Radar Techniques", TSC-PD-096-2, April 27, 1973.

7. L. E. Brennan and I. S. Reed, "Quantization Noise in Digital Moving Target Indication Systems," IEEE Trans. on Aerospace & Electronic Systems, Vol. AE5-2, No. 6, Nov. 1966.

8. L. E. Brennan, I. S. Reed, and E. Pugh, "Control Loop Noise in Adaptive Array Antennas," IEEE Trans. AES, March 1971, pp. 254-262.

9. L. E. Brennan, I. S. Reed, "Effect of Envelope Limiting in Adaptive Array Control Loops," IEEE, Vol. AES-7, July 1971.

APPENDIX I

LOSSES IN ADAPTIVE RADAR PERFORMANCE DUE TO QUANTIZATION

PRELIMINARIES

Analyses of adaptive radar performance usually neglect the detailed effects of quantization noise. If estimates of degradation in performance are made, they often depend on costly computer simulations. This situation is improved here for most cases of practical interest by developing formulas for computing the signal-to-noise loss in adpative radar performance due to quantization.

Assume the adaptive array-radar has a receiving array of M identical elements. Let $z_k(t)$ be the complex-valued process received by the k-th element for $k=1,2,\ldots,M$. In a pulse-sampling radar, pulses are transmitted periodically with pulse repetition period T. For this case the reflected signals from an object at fixed range $R = \frac{c\tau}{2}$ are proportional to the coefficient of reflectivity of that object. Here, of course, τ is the round-trip time required for a pulse to travel from the radar antenna to the object and return, and c is the velocity of light. Such a radar samples the reflectivity of the object sequentially at times $t_1,t_2,\ldots,t_L$ where $t_n = \quad . \tau$. The sampled data set associated with these sampling times is the sampled data set $\{z_k(t_n)\}$ of signals received from an object at range R. It is convenient for mathematical purposes to represent this sampled-data set as the column vector (matrix),

$$\begin{pmatrix} z_1(t_1) \\ \cdot \\ \cdot \\ \cdot \\ z_M(t_1) \\ \cdot \\ \cdot \\ \cdot \\ z_1(t_L) \\ \cdot \\ \cdot \\ \cdot \\ z_M(t_L) \end{pmatrix} \equiv \begin{pmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_M \\ \cdot \\ \cdot \\ \cdot \\ x_{(L-1)M+1} \\ \cdot \\ \cdot \\ \cdot \\ x_{ML} \end{pmatrix} \equiv X$$

where $N = M \times L$ is the number of dimensions of the vector.

Radar detection involves a choice between two hypotheses, the noise-only hypothesis H_0 and the signal-plus-noise hypothesis H_1. Assuming additive Gaussian noise of zero mean, the expected value of X, given hypothesis H_0 (noise only), is

$$EX = 0.$$

Similarly the expected value of X, given H_1 (signal plus noise) is

$$EX = S$$

whe.'e S is the column vector (matrix)

$$S = \begin{pmatrix} S_1 \\ S_2 \\ \cdot \\ \cdot \\ \cdot \\ S_N \end{pmatrix}$$

of signal echoes that might be expected from an object at range R at the M elements of the array from the L pulses. For hypothesis H_1, the noise vector is given by N = X - S. Assume that all components of M are Gaussian and jointly distributed.

In order to detect signal vector S in the received sampled-data vector X one must design a filter (a linear functional) for vector X which is tuned to signal S. Such a sampled-data filter is the scalar

$$y = \sum_{k=1}^{N} \bar{w}_k x_k = W^* X \qquad (1)$$

where W is the weight vector

$$W = \begin{pmatrix} w_1 \\ w_2 \\ \cdot \\ \vdots \\ \cdot \\ w_N \end{pmatrix}$$

of complex numbers, $\bar{w}_k$ denotes complex conjugation and W^* denotes the conjugate transpose of the column matrix W.

2.2 MOMENTS OF QUANTIZED VIDEO

To study the effects of quantization on adaptive radar performance one needs to find the moments of the sample data filter y, given in (1). To treat this problem let $[x_n] = [u_n] + i[v_n]$ be the quantized or digitalized value of the complex number $x_n = u_n + iv_n$ where $[v_n]$ denotes the digital value of the imaginary part of x_n. In the conversion from analogue to digital let

$$[x_n] = x_n + \varepsilon(x_n) \tag{2}$$

where

$$\varepsilon(x_n) = e(u_n) + i\ e(v_n) \tag{3}$$

is the complex error. Here $e(u_n)$ is the real part of $\varepsilon(x_n)$ and $e(v_n)$ is the imaginary part of error $\varepsilon(x_n)$ in the A-D conversion process. The error functions $e(u)$ and $e(v)$ are sawtooth functions of the real and imaginary parts of $x = u + iv$, respectively. Explicitly, $e(u)$ and $e(v)$ have the same Fourier series representation, given by

$$e(u) = \frac{\Delta}{\pi} \sum_{k=1}^{\infty} \frac{(-1)^k}{k} \sin\left(\frac{2\pi k u}{\Delta}\right) \text{ and} \tag{4}$$

$$e(v) = \frac{\Delta}{\pi} \sum_{k=1}^{\infty} \frac{(-1)^k}{k} \sin\left(\frac{2\pi k v}{\Delta}\right),$$

respectively, where Δ is the quantizer step size in voltage [see Reference 1].

For signal-plus-noise the first moment of $[x_n]$, the quantized version of the n-th component of the sampled-data vector X, is by (2) and (3)

$$\begin{aligned} E[x_n] &= E\Big(x_n + \varepsilon(x_n)\Big) \\ &= Ex_n + E(\varepsilon x_n) \\ &= s_n + Ee(u_n) + i\ Ee(v_n) \end{aligned}$$

Replacing $e(\sigma\ )$ and $e(\tau_n)$ by their Fourier series, eq. (4), and inverting the order of the summation and expected value operator E, yields

$$E[x_n] = s_n + \Delta \sum_{k=1}^{\infty} \frac{(-1)^k}{k} E\,\{\sin \frac{2\pi k}{\Delta} u_n + i \sin \frac{2\pi k}{\Delta} v_n\}$$

Since σ_n and τ_n are independent identically distributed Gaussian variates with means $R(s_n)$ and $I(s_n)$, respectively, where $R(s_n)$ and $I(s_n)$ are the real and imaginary components of signal component s_n,

$$E \sin\left(\frac{2\pi k}{\Delta} u_n\right) = \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{\infty} e^{-(u_n - Rs_n)^2 / 2\sigma^2} e(u_n)\,du_n \ .$$

$$= e^{-2\pi^2 k^2 (\sigma/\Delta)^2} \sin\left(\frac{2\pi k R(s_n)}{\Delta}\right) \ .$$

and

$$E \sin\left(\frac{2\pi k}{\Delta} v_n\right) = e^{-2\pi^2 k^2 (\sigma/\Delta)^2} \sin\left(\frac{2\pi k I(s_n)}{\Delta}\right)$$

Thus

$$E[x_n] = s_n + \frac{\Delta}{\pi} \sum_{k=1}^{n} \frac{(-1)^k}{k}\ e^{-2\pi k^2 (\sigma/\Delta)^2}$$

$$\times \left(\sin \frac{2\pi k R(s_n)}{\Delta} + i \sin \frac{2\pi k I(s_n)}{\Delta}\right)$$

where $R(s_n)$ and $I(s_n)$ are the real and imaginary components of the n-th component s_n of the signal vector S and σ is the standard deviation of the noise.

For most cases of practical interest in adaptive radar $\Delta << \sigma$. For this case

$$E[x_n] = s_n + O\left(e^{-2\pi^2\left(\frac{\sigma}{\Delta}\right)^2}\right)$$

where O(y) denotes the order of y as y tends to zero. Evidently $E[x_n]$ is closely approximated on the average by the s_n, i.e. $E[x_n] \approx s_n$.

The complex covariance matrix of [X], the quantized version of the data vector X for noise alone is by (3)

$$\begin{aligned} M_Q &= E[X][X]^* \\ &= E(X + \varepsilon(X))(X + \varepsilon(X))^* \\ &= EXX^* + E\varepsilon(X)\ \{\varepsilon(X)\}^* + E(\{\varepsilon(X)\}X^* + X\{\varepsilon(X)\}^*) \end{aligned} \tag{5}$$

where * denotes conjugate transpose and $\varepsilon(X)$ is the column vector of complex quantization errors $\varepsilon(x_n)$, as given by (3), namely,

$$\varepsilon(X) = \begin{pmatrix} \varepsilon(x_1) \\ \varepsilon(x_2) \\ \cdot \\ \cdot \\ \varepsilon(x_N) \end{pmatrix}$$

The first term of (5) is the covariance matrix M of the sample-data vector X, i.e.,

$$M = EXX^*$$

To treat the second term of (5) and make use of previous results, we must transform the data sample vector,

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{pmatrix} = \begin{pmatrix} u_1 + iv_1 \\ u_2 + iv_2 \\ \cdot \quad\; \cdot \\ \cdot \quad\; \cdot \\ \cdot \quad\; \cdot \\ u_N + iv_N \end{pmatrix} = V + iV \qquad (6)$$

of n complex components into the data sample vector

$$U = \begin{pmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ y_{2N} \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ u_N \\ v_1 \\ v_2 \\ \cdot \\ \cdot \\ v_N \end{pmatrix} = \begin{pmatrix} U \\ V \end{pmatrix} \qquad (7)$$

oi 2N real components where u_k is the real part of sample x_n, v_k is the imaginary part of sample x_k, and

$$U = \begin{pmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ u_N \end{pmatrix} \quad \text{and } V = \begin{pmatrix} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_N \end{pmatrix}$$

are the real and imaginary parts of X, respectively.

Let also the complex covariance matiix M and its inverse M^{-1} be decomposed into real and imaginary parts as

$$M = A + iB \quad \text{and} \quad M^{-1} = C + iD \tag{8}$$

Then the 2N × 2N symmetric matrix

$$V = \begin{pmatrix} A, & -B \\ B, & A \end{pmatrix} = EYY^T \tag{9}$$

has the inverse

$$V^{-1} = \begin{pmatrix} C, & -D \\ D, & C \end{pmatrix} \tag{10}$$

where Y^T denotes the transpose of vector Y.

Next, the quadratic form $X^*M^{-1}X$ can be shown to be identical to the quadratic form $Y^TX^{-1}Y$ where X^{-1} is given by (9). To show this

$$X^*M^{-1}X = (U + iV)\ (C + iD)\ (U + iV)$$

$$= (U - iV)^T(C + iD)\ (U + iV)$$

$$= (U - iV^T)(C + iD)\ (U + iV)$$

$$= U^TCU + V^TDU - U^TDV + V^TCV$$

where the imaginary part vanishes since M^{-1} is Hermitian and where U^T denotes the transpose of the real vector U. Similarly

$$Y^TX^{-1}Y = \begin{pmatrix} U \\ V \end{pmatrix}^T \begin{pmatrix} C, & -D \\ D, & C \end{pmatrix} \begin{pmatrix} U \\ V \end{pmatrix}$$

$$= \quad (U^TC + V^TD,\ -U^TD + V^TC) \begin{pmatrix} U \\ V \end{pmatrix}$$

$$= \quad U^TCU + V^TDU - U^TDV + V^TCV$$

Since this agrees with the above, the identity

$$X^*M^{-1}X = Y^TK^{-1}Y \tag{11}$$

is true. The joint probability density of the components of the vector X is

$$P(X) = (\pi)^{-n}|M|^{-1} \exp\ (-X^*M^{-1}X) \tag{12}$$

where $|M|$ is the determinant of the covariance matrix M. Using (11), this density is easily shown to be equivalent to the density

$$P(Y) = (\pi)^{-M}|K|^{-\frac{1}{2}}\exp\;(-Y^{*}K^{-1}Y) \qquad (13)$$

where Y is the 2n component vector, given by (7) and K^{-1} is defined by (9).

The above remarks will now be used to evaluate the second term of (5). The $(m,n)^{th}$ element of this is

$$\begin{aligned} E\left(\varepsilon(x_m)\overline{\varepsilon(X_n)}\right) &= E\left(e(u_m)+ ie(v_m)\right)\left(e(u_n) - ie(v_n)\right) \\ &= E\; e(u_m)e(u_n) + E\; e(v_m)e(v_n) \\ &\quad + i\left(E\; e(u_m)e(v_n) - E\; e(u_n)e(v_n)\right) \\ &= 2\left(E\; e(u_m)e(u_n) + i\; E\; e(u_m)e(v_n)\right) \qquad (14) \end{aligned}$$

where the last line follows from (9). The first term of (14) is

$$E\; e(u_m)e(u_n) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(u_m,u_n)e(u_m)e(u_n)du_m du_n$$

$$= \left(\frac{\Delta}{\pi}\right)^2 \sum_{k=1}^{\infty}\sum_{\ell=1}^{\infty} \frac{(-1)^{k+\ell}}{k\ell}\; I_{k\ell}$$

where

$$Ik\ell = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-(u_m^2 + u_n^2 - 2\rho_{mn}u_m u_n)/2\ {}^2(1-\rho_{mn}^2)} \sin\left(\frac{2\pi k}{\Delta}\, u_m\right)$$

$$+ \sin\left(\frac{2\pi\rho}{\Delta}\, u_n\right)\frac{du_m du_n}{2\pi\sigma\sqrt{1-\rho_{mn}^2}} \quad ,$$

$$\sigma^2\rho_{mn} = E(u_m u_n) = E(v_m v_n)$$

and

$$\rho_{mm} = \rho_{nn} = 1.$$

The above integral can be evaluated to yield finally

$$E\{e(u_m)e(u_n)\}$$

$$= \frac{\Delta^2}{\pi^2} \sum_{K}\sum_{\ell=1}^{\infty} \frac{(-1)^{K+\ell}}{k\ell}\, e^{-2\pi^2\left(\frac{\sigma}{\Delta}\right)^2 (K^2+\ell^2)} \sin\{4\pi^2\left(\frac{\sigma}{\Delta}\right)^2 K\ell\rho_{mn}\} \qquad (15)$$

Again for the cases of interest $\Delta << \sigma$ so that to first order in $e^{-2\pi^2(\sigma/\Delta)^2}$ only the $k=\ell$ terms need be kept. Thus

$$E\{e(u_m)e(u_n) = \frac{\Delta^2}{2\pi^2} \sum_{K=1}^{\infty} \frac{1}{K^2}\, e^{-4\pi^2\left(\frac{\sigma}{\Delta}\right)^2 K^2(1-\rho_{mn})} \qquad (16)$$

$$+ 0\ (e^{-2\pi^2(\sigma/\Delta)^2}$$

Finally if $\frac{\Delta^2}{12}$, the quantization noise power, is set below the natural receiver noise of the radar, only the terms in (16) for which m=n will persist. To this final approximation

$$E\ e^2(u_n) = \frac{\Delta^2}{2\pi^2}\sum_{K=1}^{\infty}\frac{1}{k^2} + 0\left(e^{-2\pi^2\ (\sigma/\Delta)^2}\right) \tag{17}$$

$$= \frac{\Delta^2}{12} + 0\left(e^{-2\pi^2(\sigma/\Delta)^2}\right)$$

and

$$Ee(u_m)e(u_n) = 0 + 0\left(e^{-2\pi^2(\sigma/\Delta)^2(1-\rho_{mn})}\right), \quad m\neq n,$$

using the well-known identity,

$$\sum_{K=1}^{\infty}\frac{1}{K^2} = \frac{\pi^2}{6}$$

If the quantization noise power is again set below the receiver noise, the imaginary part can be estimated in a similar manner to be

$$Ee(u_n)e(v_n) = 0 + 0(e^{-2\pi^2(\sigma/\Delta)^2(1-\mu_{mn})} \tag{18}$$

for all m and n where

$$\sigma^2\ \mu_{mn} = E\ u_m v_n = -E\ v_n u_m$$

In this case $\mu_{mn} = 0$, since by (7), (8) and (9),

$$B = EV^T = -EUV^T = -B^T$$

so that B is a skew symmetric matrix. Combining (17) and (18) yields

$$E\, \varepsilon(X)\{\varepsilon(X)\}^* = \frac{\Delta^2}{6} I \tag{19}$$

where I is the N x N identity matrix as an estimate of the second term of (5), assuming the quantization noise is comparable to the receiver noise power.

A similar analysis will show (see reference 1) that the matrix elements of the third term of (5) are zero to the order of $e^{-2\pi^2(\sigma/\Delta)^2}$. Combining this with (19), yields finally,

$$M_Q = E[X][X]^* = M + \frac{\Delta^2}{6} I \tag{20}$$

to the order of $e^{-2\pi^2(\sigma/\Delta)^2(1-\rho_{mn})}$. This result will now be used to estimate losses in detection sensitivity an adaptive radar will suffer as a function of the quantization step Δ. Better approximations to M_Q, then given by (20), particularly for a larger quantization step Δ, will be the subject of a future study.

SIGNAL-TO-NOISE LOSS DUE TO QUANTIZATION

A best sampled-data filter of form (1) is one in which the weight vector W is chosen to maximize the signal-to-noise (S/N) ratio. The weight vector which achieves the optimum S/N ratio is well known (see for example Reference 2) and given by

$$W_o = M^{-1}S \tag{21}$$

where M is the covariance matrix for noise only. Since M is often not known a priori, an estimate $\hat{M}$ is used instead, depending on K samples of the noise process. One estimate is the sample average. If $X^{(1)}, X^{(2)}, \ldots X^{(K)}$ are K independent samples of the noise process, then the sample average estimate of M is given by

$$\hat{M} = \frac{1}{K}\sum_{j=1}^{K} X^{(j)}X^{(j)^*} \tag{22}$$

If $\hat{M}$ is used in (21), then

$$\hat{W} = \hat{M}^{-1}S \tag{23}$$

is a near optimum set of weights to be used in filter (1). The output of this filter is

$$\hat{u} = \hat{W}^*X \tag{24}$$

where $\hat{W}$ is given by (23). The output signal-to-noise ratio conditioned on a knowledge of $\hat{W}$, is

$$(S/N|\hat{W})_o = \frac{(E\hat{u}|\hat{w}))^2}{\mathrm{Var}(\hat{u}|\hat{w})}$$

$$= (S^*\hat{M}^{-1}S)^2/S^*\hat{M}^{-1}M\hat{M}^{-1}S \tag{25}$$

Previously it was shown [2] that if this signal-to-noise ratio was normalized with respect to its maximum value $S^{*}M^{-1}S$ that this quantity, namely,

$$\rho(\hat{M}) = (S/N|\hat{w})_o \, S^{*}M^{-1}S \tag{26}$$

was a random variable which was in the interval $0 \leq \rho(\hat{M}) \leq 1$. It was further found that this normalized signal-to-noise ratio has a probability density $P(\rho)$ which depedned only on N, the number of components of X, and the number K of sample vectors.

Previously in Reference 2 the quantity $\rho(\hat{M})$ was the (S/N) loss ratio which depended on $\hat{M}$, the estimate of M, without any assumption of quantization. To make this ratio depend on quantization let $\hat{M}_Q$ be the sample average estimate of M, including quantization, i.e. let

$$\hat{M}_Q = \frac{1}{K} \sum_{j=1}^{K} [X^{(j)}][X^{(j)}]^{*} \tag{27}$$

where $[X^{(j)}]$ denotes the quantized value of sample vector $X^{(j)}$. Then the normalized signal-to-noise ratio for this estimate of M is

$$\rho_{Total}(\hat{M}_Q) = \frac{(S/N|\hat{M}_Q)_o}{S^{*}M^{-1}S}$$

$$= \frac{(S/N|M_Q)_o}{S^{*}M_Q^{-1}S} \left(\frac{S^{*}M_Q^{-1}S}{S^{*}M^{-1}S} \right)$$

$$= \rho(\hat{M}_Q)\left(\frac{S^*M_Q^{-1}S}{S^*M^{-1}S}\right) \qquad (28)$$

where

$$M_Q = E[X][X]^*$$

is the complex covariance matrix of [X], given by (5) and (20). For K reasonably large the statistics of [X] will be close to Gaussian, hence the statistics of $\rho(\hat{M}_Q)$ will closely approximate the statistics of $\rho(\hat{M})$ as given by (26). Thus, taking the expected value of (28), yields by equation (18) of Reference (2),

$$E\rho_{Total}(\hat{M}_Q) = \left(\frac{K+2-N}{K+1}\right)\left(\frac{S^*M_Q^{-1}S}{S^*M^{-1}S}\right) \qquad (29)$$

This is the expected loss in S/N ratio, due firstly to the fact that only a finite number K of samples were used to estimate M, and secondly to the sample quantization. Expressed in decibels, this expected loss ratio is

$$\text{Loss} = -10 \log_{10}\{(K+2-N)/(K+1)\}$$
$$-10 \log_{10}\{(S^*M_Q^{-1}S)/(S^*M^{-1}S)\} \qquad (30)$$

where the first term is the loss in decibles, due to the finite sampling in estimating M, and the second term is the loss due to quantization of finite step size Δ. Using (20), an estimate of the last term can be made from the identity,

$$M_Q^{-1} = M^{-1}(I+\alpha M^{-1})^{-1} = M^{-1}(I-\alpha M^{-1}) + 0(\alpha^2)$$

$$\approx M^{-1} - \alpha M^{-2}$$

where

$$\alpha = \Delta^2/6.$$

Hence

$$S^* M_Q^{-1} S = S^* M^{-1} S - \frac{\Delta^2}{6} S^* M^{-2} S$$

and

$$\text{Loss}_Q = -10 \log_{10} \left(1 - \frac{\Delta^2}{6} \frac{(M^{-1}S)^*(M^{-1}S)}{S^* M^{-1} S}\right)$$

$$= -10 \log_{10} \left(1 - \frac{\Delta^2}{6} \frac{W_o^* W_o}{S^* M^{-1} S}\right) \qquad (31)$$

is the S/N loss due to quantization where W_o is the optimum weight vector (21). The reliability of these loss formulas will be checked by simulation during the next quarter.

REFERENCES

1. L. E. Brennan and I. S. Reed, "Quantization Noise in Digital Moving Target Indication Systems," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-2, No. 6, Nov. 1966.

2. I. S. Reed, J. D. Mallett, L. E. Brennan, "Rapid Convergence Rate in Adaptive Arrays," IEEE Trans. of Aerospace & Electronic Systems, Vol. AES-10, No. 6, Nov. 1974.

APPENDIX II

FORTRAN PROGRAM FOR SIMULATION OF ADAPTIVE AMTI WITH NON-LINEARITIES

```
      PROGRAM ADAPTM8
C
C    MOD INCLUDES SAMPLE MATRIX INVERSION SMI,INVERSE MATRIX UPDATE
C    IMU, AND APPLEBAUM LOOPS
C    NP=NUMBER OF PULSES
C    NEL=NUMBER OF ANTENNA ARRAY ELEMENTS
C    NSC=NUMBER OF CLUTTER SCATTERERS PER RANGE RING
C    ALL DISTANCES ARE MEASURED IN WAVELENGTHS
C    D=DISTANCE RADAR PLATFORM MOVES BETWEEN PULSES
C    ELSPAC=ELEMENT SPACING
C    THETAD=INITIAL SCAN ANGLE IN DEGREES,MEASURED FROM GROUND TRACK
C    THDOTD=SCAN RATE IN DEGREES PER PULSE REPETITION PERIOD
C    RECN=RATIO OF RECEIVER NOISE TO CLUTTER POWERS
C    SIG=AMPLITUDE OF STEERING SIGNALS IN LOOPS
C    XR=ANY RANDOM NUMBER TO INITIALIZE THE CLUTTER AND NOISE GENERATORS
C    IELPAT CONTROLS THE ELEMENT PATTERN***IELPAT=1 FOR APS/96 ELEMENT PA
C           OTHERWISE THE ELEMENT PATTERN IS ISOTROPIC
C    IEIG=1 NORMAL COORDINATES USED IN SIMULATION
C    NPR=SAMPLES IN ONE GROUP
C    NBP=NUMBER OF GROUPS
C    AL=WEIGHTING OF NEW DATA FOR IMU
C      DLSG = DEL/SG
C      ITL SELECIS CASE    NSC1 VALUES OF DLSG OR BITS
C                          NSC2 VALUES OF AK (AK * SG = SAT)
C      ITL = 0  LIMITING ONLY NSC1=1, DLSG=0, NSC2 VALUES OF AK
C      ITL = 1 QUANT ONLY NSC1 VALUES OF DLSG, NSC2=1, AK .GE. 5
C      ITL = 3  QUANT AND LIMITING NSC1 VALUES OF NUMBER OF HITS AND NSC2
C                          VALUES OF AK FOR EACH ABIT VALUE
      CALL MAIN
      STOP
      END


      SUBROUTINE MAIN
      COMPLEX W(30),WO(30),SCAT(60),SST(30),DCT(30)
      COMPLEX GAIN(30,60),VS(8,320), SC(30,30),CM(30,30),V(8,320)
      COMPLEX CN(30,30),A(30,30),B(30,30),CO(30,30),CC,SS
      REAL LSS, LTOT
      DIMENSION EIG(30),SEIG(30),CAS1(20),CAS2(20)
      DIMENSION Y11(500),Y12(500),Y21(500),Y22(500),Y31(500),Y32(500)
      DIMENSIONX1(500),SCL(500),X2(100),SCS(100),X3(100),SCI(100)
      COMMON/PARAM/NP,NEL,NPL,NSC,SIG
      COMMON/ARRAY/GAIN,VS,SST
      COMMON/B/W,WO
      COMMON/M/SC
      COMMON/N/CM
      COMMON/P/X1,X2,X3,SCL,SCS;SCI,I1,I2,I3,INV,LP,ISP,NRS
      COMMON/Q/CO
      COMMON/R/DCT,THDCT,ADCT,JDCT
      COMMON/NS/ICX,ICY,ICT
      COMMON/Y/Y11,Y12,Y21,Y22,Y31,Y32
      NAMELIST/INPUT/ NP,NEL,NSC,NRUNS,NSIC,NPR,INV,LP,ISP,NRS,
     1ERR,GLOP,AL,NBP,D,SIG,XR,IELPAT,ELSPAC,THETAD,THDOTD,RECN,
     2IEIG,IPLOT,ASIG,ADCT,THDCTD,JDCT,KDCT,AK,NBIT,DEL,SAT,DLSG
     3,TLDB,NRNS,NCS1,NCS2,ITL,ILT
      READ(5,INPUT)
      IF(NEL.EQ.0) RETURN
      ISAV=IEIG
      PI=4.*ATAN(1.)
      RAD=PI/180.
      XR=RANF(XR)
      NRUNS=NBP*NPR
 1    FORMAT(10F5.1)
      THDOT=THDOTD*RAD
      THETA=THETAD*RAD
      THDCT=THDCTD*RAD
      NPL=NP*NEL
      WRITF(6,INPUT)
C   IDEAL PERFORMANCE AT ANGLE THETA
      NR=1
      CALL PHASE       (D,CONST,SCZERO,IELPAT,ELSPAC,THETA,THDOT, 0,NR)
      WRITE(6,542) CONST,SCZERO
      SRECN=RECN*NEL**3/2./SCZERO*SIG**2
      VRECN=SQRT(SRECN)
C     IDEAL COV. MATRIX
      CALL         CVM(NPL,SC,NSC,SRECN)
      TAU=GLOP*REAL(CM(1,1))*NPL/2./ERR
      PRINT 500,TAU
      PRINT 500,( SC(I,I)          ,I=1,NPL)
      CC=0.
      DO 12 I=1,NPL
 12   CC=CC+SC(I,I)
      CC=CC/NPL/2.
      SG=SQRT(CABS(CC))
      PRINT 530,SG
      PRINT 510
      PRINT 502
      CALL WTS(W,SC,CONST,0,0)
C     SC = INVERSE IDEAL MATRIX
      NRS=4$I3=1
      CALL SIGCN(W,SST,NPL,CONST)
      NRS=3
      CALL SIGCN(W,SST,NPL,CONST)
      S1=SCI(1)
      SCI(1)=10.*ALOG10(SCI(1))
      SSG=SCI(1)
      SCS(1)=SSG
C     DIAGONAL OF SC
      PRINT 510
      PRINT 500,(SC(I,I),I=1,NPL)
C COMP VS FOR EACH SAMPLE
      AD=ADCT*NPR*NBP
      NBPR = NPR*NBP
      DO 31 J = 1,NBPR
      CALL SCGEN(NSC,SCAT,XR)
      DO 31 L=1,NPL
      CC=0.
      IF(J.EQ.NSIC) CC = ASIG*CONJG(SST(L))
      IF(J.EQ.JDCT) CC = CC+AD*DCT(L)
      DO 32 K=1,NSC
      CC=CC+SCAT(K)*GAIN(L,K)
 32   CONTINUE
   31 VS(L,J)=CC
      IF(RECN.GT.0.) CALL NADD(J,NBPR,VRECN,XR)
      DO 37 J=1,NBPR
      DO 37 L=1,NPL
   37 V(L,J)=VS(L,J)
      CST=0.
      DO 20 J=1,NBPR
      DO 20 L=1,NPL
 20   CST=CST+REAL(V (L,J)*CONJG(V (L,J)))
      CST=CST/2.
      SG=SQRT(CST/(NBPR*NPL))
      PRINT 530,SG
      SS = 0.
      CC = 0.
      DO 10 I = 1,NPL
      SS = SS+W(I)*CONJG(W(I))
   10 CC = CC+SST(I)*CONJG(W(I))
      AC=REAL(CC/SS)
      XX=S1/CONST
      PRINT 500,AC,XX
C
C
C
 16   CONTINUE
      NBPR=NPR*NBP
      LSS=10.*ALOG10((NPR+2.-NPL)/(NPR+1.))
      NB=NBP+1
      DO 94 I=1,NB
 94   SCI(I)=SSG
      PRINT 509,ITL,ILT
      READ(5,1) (CAS1(I),I=1,NCS1)
      READ(5,1) (CAS2(I),I=1,NCS2)
      DO 301 IS=1,NCS1
      DLSG=CAS1(IS)
      DO 300 IR=1,NCS2
      ICX=ICY=ICT=0
      PRINT 510
      PRINT 510
      PRINT 500,CAS1(IS),CAS2(IR)
      AK=CAS2(IR)
      DO 35 M = 1,NPL
      DO 35 N = 1,NBPR
   35 VS(M,N) = V(M,N)
      IF(ITL.EQ.0) GO TO 22
      IF(ITL.EQ.1) GO TO 4
      IF(ITL.EQ.3)GO TO 3
      TLDB=CAS1(IS)
      TLOS=10.**(-CAS1(IS)/10.)
      DLSG=SQRT((1.-TLOS)*AC*6.)/SG
 4    CONTINUE
      IF(DLSG.LE. 0.) GO TO 91
      NL=2.*AK /DLSG+.5
      NBIT=ALOG(NL+2.)/ALOG(2.)+.5
      AK=DLSG*NL/2.
      GO TO 5
 3    NBIT=CAS1(IS)+.5
      NL=2**NBIT-2
      DLSG=2.*AK/NL
 5    CONTINUE
      DS1=(DLSG*SG)**2/6.
      DO 14 M=1,NPL
      DO 14 N=1,NPL
      CC      =CM(M,N)
      IF(M.EQ.N) CC=CC    +DS1
 14   SC(M,N)=CC
      CALL MATINV(NPL,SC)
      CC=0.
      DO 11 M=1,NPL
      SS=0.
      DO 13 N=1,NPL
 13   SS=SS+SC(M,N)*SST(N)
 11   CC=CC+CONJG(SST(M))*SS
      SSC=REAL(CC)*CONST
      XX=SSC
      TLD1=10.*ALOG10(SSC)-SC1(1)
      LTOT=LSS+TLD1
      PRINT 525,TLD1,LSS,LTOT
      XX=1.-DS1/AC
      IF(XX.GT.0.) DB = 10.*ALOG10(ABS(XX))
      PRINT 500,XX,DB
      ABIT=NBIT
      ANL=NL
   22 CONTINUE
      SAT=SG*AK
      PRINT 526
      PRINT 500, SAT,AK,TLDB,DLSG,ABIT,AC,CC,SS,ANL
      IF(ILT.EQ.1) CALL QUANT1(VS,NL,SAT,NPL,NBPR,ITL)
      IF(ILT.EQ.2) CALL QUANT2(VS,NL,SAT,NPL,NBPR,ITL)
      CT=FLOAT(ICT)
      CX=FLOAT(ICX)
      CY=FLOAT(ICY)
      PRINT 514
      PC=(CX+CY)/CT*100.
      PRINT 516,PC
      PRINT500,CT,CX,CY
      PRINT 510
 91    CONTINUE
       I1=1$I2=1$I3=1
       X1(1)=0$X2(1)=0$X3(1)=0
       NPL4=4*NPL
C
C     BIG LOOP
C     SIMULATION USING RANDOM CLUTTER SAMPLES
       NBO=1
       NCT=0
       JJ=0
       DO 87 I = 1,NBP
       JJ = JJ+NPR




C     SMI
       NRS=2
       I2=I2+1
       X2(I2)=JJ
       N1=JJ-NPR+1
       CALL SAMPLE(NPL,NPR,I,CONST,N1)
  2    CONTINUE
 87    CONTINUE
C
       AV = 0.
       AVS=0.
       VA = 0.
       IZ=I2-1
       DO 28 I = 2,IZ
       AVS=AVS+SCS(I)
    28 AV = AV+SCI(I)
       AV=AV/(NBP-1)
       AVS=AVS/(NBP-1)
       VA=0.$VAS=0.
       DO 29 I = 2,I2
       IF(I.EQ.I3) GO TO 90
       VAS=VAS+(SCS(I)/AVS-1.)**2
       VA =VA +(SCI(I)/AV -1.)**2
 90    SCS(I)=10.*ALOG10(SCS(I))
 29    SCI(I)=10.*ALOG10(SCI(I))
       VA=VA/(NBP-1)
       VAS=VAS/(NBP-1)
       AV =10.*ALOG10(AV )-SCI(1)
       AVS=10.*ALOG10(AVS)-SCI(1)
       PRINT 504
       PRINT 500,(X2(I),I=1,I2)
       PRINT 506
       PRINT 515,AVS,VAS
       PRINT 500,(SCS(I),I=1,I2)
       AVL=0.
       VAL=0.
```

```
      DO 30 I=3,I2
30    AVL=AVL+SCL(I)
      AVL=AVL/(NBP-1)
      DO 36 I=3,I2
      VAL=VAL+(SCL(I)/AVL-1.)**2
36    SCL(I)=10.*ALOG10(SCL(I))
      VAL=VAL/(NBP-1)
      AVL=10.*ALOG10(AVL)-SCI(1)
      PRINT 515,AVL,VAL
      PRINT 500,(SCL(I),I=3,I2)
      PRINT 510
      IF(INV.EQ.1)
     1PRINT 507
      IF(INV.EQ.1)
     2PRINT 500,AV,VA
      IF(INV.EQ.1)
     3PRINT 500,(SCI(I),I=1,I3)
      PRINT 510
      GO TO (80,81,82), IS
   80 Y11(IR)=SSG+AVS
      Y12(IR)=SSG+AVL
      GO TO 300
   81 Y21(IR)=SSG+AVS
      Y22(IR)=SSG+AVL
      GO TO 300
   82 Y31(IR)=SSG+AVS
      Y32(IR)=SSG+AVL
300   CONTINUE
301   CONTINUE
      DO 302 I = 1,NCS2
  302 X1(I)=CAS2(I)
      IF(IPLOT.EQ.1) CALL BPLOT(NCS2,SSG)
      XR=RANF(-1.)
      AK=AKS
      READ INPUT
      IF(NEL.EQ.0) RETURN
      XR=RANF(XR)
      GO TO 16
500   FORMAT(1X,8E16.7)
501   FORMAT(/,*  NUMBER OF SAMPLES = *,I5)
502   FORMAT(/ ,* OPT PERFORMANCE*)
503   FORMAT(/,* XR= *,F17.10,* TAU= *,E17.10)
504   FORMAT(*   X AXIS*)
506   FORMAT(*   SAMPLE MATRIX INVERSION(SMI,SCS)*)
507   FORMAT(*   INVERSE MATRIX UPDATE(IMV,SCI)*)
508   FORMAT(1X,10I4)
509   FORMAT(/,*          ITL= *,I4,* ILT= *,I4)
510   FORMAT(/)
 511  FORMAT(/,* EIGEN VALUES*)
 512  FORMAT(/,* IDEAL COVAR. MATRIX*)
 513  FORMAT(/,* NORMAL MATRIX*)
514   FORMAT(1X,*  CTOT  CX  CY*)
```

```
 515 FORMAT(/,1X,*SIMULATION LOSS = *,F10.3,*VARIANCE = *,F10.3)
 516 FORMAT(/,1X,*PERCENT LIMITING = *,F10.3)
 525 FORMAT(/,1X,*TH. QUANT LOSS DB = *,F10.3,*TH. SAMPLE LOSS = *,F10.
    13,*TOTAL TH. LOSS = *,F10.3)
526  FORMAT(/,*  SAT   AK   TLDB    DLSG    ABIT    AC   SC   SS*)
527  FORMAT(1X,10,F12.3)
530  FORMAT(/,*  SG= *,2E16.7)
 542 FORMAT(7H CONST=,E16.6,10X,7HSCZERO=,E16.6//)
     END


     SUBROUTINE BPLOT(NSP,SSG)
     COMMON /PARAM/ NP,NEL,NPL,NSC,SIG
     DIMENSION Y11(500),Y12(500),Y21(500),Y22(500),Y31(500),Y32(500)
     DIMENSIONX1(500),SCL(500),X2(100),SCS(100),X3(100),SCI(100)
     DIMENSION AM(500)
     COMMON/P/X1,X2,X3,SCL,SCS,SCI,I1,I2,I3,INV,LP,ISP,NRS
     COMMON/Y/Y11,Y12,Y21,Y22,Y31,Y32
     DO 3 I = 1,NSP
   3 AM(I) = SSG
     NS1=NSP+1
     NS2=NSP+2
     X1(NS1)=2.
     X1(NS2)=1.0
     Y11(NS1)=10.
     Y12(NS1)=10.
     Y21(NS1)=10.
     Y22(NS1)=10.
     Y31(NS1)=10.
     Y32(NS1)=10.
     Y11(NS2)=5.
     Y12(NS2)=5.
     Y21(NS2)=5.
     Y22(NS2)=5.
     Y31(NS2)=5.
     Y32(NS2)=5.
     AM(NS1)=10.
     AM(NS2)=5.
     CALL PLOTTSC(7HMALLETT,1.)
     CALL AXIS(0.,0.,7HSAT/SIG,-7,6.,0.,2.,1.0,0)
     CALL AXIS(0.,0.,12HMTI GAIN(DB),12,8.,90.,10.,5.,-1)
     HT=(SSG-Y32(NS1))/Y32(NS2)+0.1
     CALL SYMBOL(3.0,HT ,.1,23HSTEADY STATE GAIN(DB) =,0.,23)
     CALL NUMBER(5.0,HT ,.1,SSG,0.,4HF6.1)
     CALL LINE(X1,AM,NSP,1,0,0)
     CALL LINE(X1,Y11,NSP,1,1,1)
     CALL LINE(X1,Y12,NSP,1,1,2)
     CALL LINE(X1,Y21,NSP,1,1,1)
     CALL LINE(X1,Y22,NSP,1,1,2)
     CALL LINE(X1,Y31,NSP,1,1,1)
     CALL LINE(X1,Y32,NSP,1,1,2)
     CALL PLOT(10.,0.,-3)
```

```
      RETURN
      END

      SUBROUTINE PHASE(D,CONST,SCZERO,IELPAT,ELSPAC,THETA ,THDOT ,NC,NR)
C     COMPUTES RETURN FROM EACH SCATTER TO EACH ELEMENT
      COMMON/PARAM/NP,NEL,NPL,NSC,SIG
      COMMON/ARRAY/GAIN,VS,SST
      COMMON/K/DCT,THDCT,ADCT,JDCT
      COMPLEX GAIN(30,60),VS(8,320),DCT(30),SST(30),CP
      PI=4.*ATAN(1.)
      CP=CMPLX(0.,2.*PI)
      CZERO=0.
      THMIN=THETA-PI/2.
      THMAX=THETA+PI/2.
      C=(THMAX-THMIN)/NSC
      KK=INT((THDCT-THMIN)/C+1)
      THDCT=THMIN+(KK-.5)*C
      NS=NSC+INT((NR*THDOT)/C) +1
      PSI=THETA+NC*THDOT
      DO 10 K=1,NS
      TH=THMIN+(K-.5)*C
      CC=COS(TH)
      DO 10 M=1,NP
      MNM=M-1+NC
      SS=PSI-TH
      IF(ABS(SS).GT.PI/2.) GO TO 10
      SS=SIN(SS)
      GT=NEL
      IF(SS.EQ.0) GO TO 88
      GT=SIN(PI*ELSPAC*SS*NEL)/SIN(PI*ELSPAC*SS)
   88 CONTINUE
      IF(IELPAT.NE.1) GO TO 30
      GAM=.207*COS(PSI-TH)-.238
      ELP=SIN(21.*PI*GAM)/SIN(PI*GAM)
      GT=GT*ELP**2
   30 CONTINUE
      IF(M.EQ.1) CZERO=CZERO+    GT**4
      DO 10 N=1,NEL
      L=(M-1)*NEL+N
      GAIN(L  ,K)=GT*CEXP(-CP*((N-(NEL+1)/2.)*SS*ELSPAC-MNM  *2.*D*CC))
      IF(K.EQ.KK) DCT(L)=GAIN(L,K)
 10   CONTINUE
      SCZERO=NEL**4*SIG**2/CZERO
      CONST=NEL**2/SCZERO
      IF(JDCT.EQ.0) GO TO31
      CC=COS(THDCT)
      SS=PSI-THDCT
      GT=0.
      IF(ABS(SS).GT.PI/2.) GO TO 91
      SS=SIN(SS)
      GT=NEL
      IF(SS.EQ.0) GO TO 91
      GT=SIN(PI*ELSPAC*SS*NEL)/SIN(PI*ELSPAC*SS)
   91 CONTINUE
      IF(IELPAT.NE.1) GO TO 31
      TH=THDCT
      GAM=.207*COS(PSI-TH)-.238
      ELP=SIN(21.*PI*GAM)/SIN(PI*GAM)
      GT=GT*ELP**2
   31 CONTINUE
      DO 20 M=1,NP
      L=(M-1)*NEL+1
      MNM=NC+M-2
      SST(L  )=SIG*CEXP(-CP*( MNM*2.*D*COS(PSI ) +(M-1)/2.))
      DO 20 N=1,NEL
      LL=L+N
      IF(JDCT.EQ.0) GO TO 92
      I=L+N-1
      DCT(I)=      GT*CEXP(-CP*((N-(NEL+1)/2.)*SS*ELSPAC-MNM  *2.*D*CC))
   92 CONTINUE
   20 SST(LL )=SST(L)
      RETURN
      END


      SUBROUTINE NADD(J,NPR,    RECN,XR)
C    REC. NOISE GEN.
      COMPLEX SST(30),GAIN(30,60),VS(8,320),CP
      COMMON/PARAM/NP,NEL,NPL,NSC,SIG
      COMMON/ARRAY/GAIN,VS,SST
      CP=CMPLX(0.,2.*3.14159265)
      DO 10 J=1,NPR
      DO 10 M=1,NPL
      XR=RANF(0.)
      AMP=RECN*SQRT(-    ALOG(XR))
      XR=RANF(0.)
   10 VS(M,J)=VS(M,J)+AMP*CEXP(CP*XR)
      RETURN
      END


      SUBROUTINE WTS(W,CM,CONST,J1,J2)
C    OPTIMUM WEIGHTS
      COMPLEX SST(30),W(30),CFPREC,C,S
      COMPLEX CM(30,30),GAIN(30,60),VS(8,320)
      COMMON/PARAM/NP,NEL,NPL,NSC,SIG
      COMMON/ARRAY/GAIN,VS,SST
      S=0.
      NMAT=NP*NEL
      IF(J1.EQ.0) CALL MATINV(NMAT,CM)
      IF(J2.NE.2) PRINT 91
   91 FORMAT(/7X,6HPULSE ,4X,7HELEMENT,6X,4HREAL,5X,9HIMAGINARY,5X,
     19HAMPLITUDE,5X,5HPHASE//)
```

```
      DO 90 M=1,NP
      J=(M-1)*NEL
      DO 90 N=1,NEL
      MM=N+J
      C=0.
      DO 31 M1=1,NP
      I=(M1-1)*NEL
      DO 31 N1=1,NEL
      NN=N1+I
      C=C+SST(NN)*CM(MM,NN)
31    CONTINUE
      S=S+C*CONJG(SST(MM ))
      W(MM )=C
      IF(J2.EQ.2) GO TO 90
      AMP=CABS(C)
      PHA=ATAN2(AIMAG(C)      ,REAL(C))
      WRITE(6,50) M,N,C      ,AMP,PHA
 90   CONTINUE
      IF(J2.NE.0) RETURN
   50 FORMAT(2I11,5X,2F20.9,2F12.7)
      SR=CABS(S)
      IF(SR.LT.10.E-10) GO TO 2
      SSR=SR
      SR=10.*ALOG10(SR)
      CST=10.*ALOG10(CONST)
      GO TO 3
  2   CONTINUE
      SRX=0.
 3    SRX=SR+CST
      PRINT 52,SR,SRX
 52    FORMAT(//5X,7HWS(DB)=,G15.8,14HS/C*CONST(DB)=,G15.8)
      RETURN
      END


      SUBROUTINE SCGEN(NSC,SCAT,XR)
C   NOISE GEN. FOR EACH SCATTERER
      COMPLEX CP,SCAT(60)
      CP=CMPLX(0.,2.*3.14159265)
      DO 10 N=1,NSC
      AMP=RANF(0.)
      XR=RANF(0.)
   10 SCAT(N)=SQRT(-   ALOG(AMP))*CEXP(CP*XR)
      RETURN
      END


      SUBROUTINE SIGCN(W,SST,NPL,CONST)
C COMPUTE SIG/ CLUT USING WTS ON IDEAL MATRIX
      COMMON/N/CM
      COMMON/P/X1,X2,X3,SCL,SCS,SCI,I1,I2,I3,INV,LP,ISP,NRS
      DIMENSIONX1(500),SCL(500),X2(100),SCS(100),X3(100),SCI(100)
      COMPLEX CM(30,30),W(30),SST(30),CC,SS,CB
      SS=0.
      CC=0.
      DO 71 M=1,NPL
      SS=SS+W(M)*CONJG(SST(M))
      CB=0.
      DO 70 N=1,NPL
 70   CB=CB+            W(N)*CM(M,N)
 71   CC=CC+CB*CONJG(W(M))
      SSR=CABS(SS)**2
      CCR=CABS(CC)
      SCD=SSR/CCR*CONST
      SCR=SCD
      IF(NRS.EQ.1) SCL(I1)=SCR
      IF(NRS.EQ.3) SCI(I3)=SCR
      IF(NRS.EQ.2) SCS(I2)=SCR
      IF(NRS.NE.4) GO TO 2
      SCDB=10.*ALOG10(SCD)
      SSR=10.*ALOG10(SSR)
      CCR=10.*ALOG10(CCR)
      WRITE(6,80) SSR,CCR,SCDB
   80 FORMAT(//8H SIGNAL=,F10.5,4X,14HCLUTTER POWER=,F10.5,4X,8HS/C(DB)
     1,F10.5)
 2    CONTINUE
      RETURN
      END


      SUBROUTINE CVM(NPL,SC,NSC,SRECN)
C    FORMS STEADY STATE COV. MATRIX
      COMPLEX SST(30),DCT(30),CC
      COMPLEX CM(30,30),SC(30,30),GAIN(30,60),VS(8,320)
      COMMON/ARRAY/GAIN,VS,SST
      COMMON/N/CM
      COMMON/K/DCT,THDCT,ADCT,JDCT
      A=ADCT*ADCT
      DO 10 M=1,NPL
      DO 10 N=M,NPL
      CC=0.
      IF(JDCT.NE.0) CC=CC+A*DCT(N)*CONJG(DCT(M))
      DO 21 J=1,NSC
 21   CC=CC+GAIN(N,J)*CONJG(GAIN(M,J))
      IF(M.EQ.N) CC=CC+SRECN
      CM(M,N)=CC$CM(N,M)=CONJG(CC)
      SC(M,N)=CC$SC(N,M)=CONJG(CC)
 10   CONTINUE
      RETURN
      END


      SUBROUTINE SAMPLE(NPL,NPR,I,CONST,N1)
      COMPLEX W(30),WO(30),SST(30),CC
      COMPLEX CO(30,30),CN(30,30),CZ(30,30),GAIN(30,60),VS(8,300)
      COMMON/Q/CO
      COMMON/B/W,WO
      COMMON/ARRAY/GAIN,VS,SST
C COMPUTE SAMPLE COV. MATRIX


      N2=N1+NPR-1
       A=1./I
      IF(I.NE.1) GO TO 2
      A=1.
      DO 8 M=1,NPL
      DO 8 N=1,NPL
 8    CN(M,N)=0.
 2    CONTINUE
      IF(A.GT.1.) A=1.
      DO 11 M=1,NPL
      DO 11 N=M,NPL
      CC=0.
      DO 30 K=N1,N2
      CC=CC+CONJG(VS(M,K))*VS(N,K)
30    CONTINUE
      CN(M,N)=CC/NPR
      CZ(M,N)=CN(M,N)
      CN(N,M)=CONJG(CN(M,N))
      CZ(N,M)=CN(N,M)
 11   CONTINUE
 501  FORMAT(1X,* SAMPLE INVERSION*)
      CALL WTS(W,CN,CONST,0,2)
      CALL       SIGC2(W,SST,NPL,CONST,I,CZ)
 3    CONTINUE
      RETURN
      END


      SUBROUTINE SIGC2(W,SST,NPL,CONST,I,UM)
      COMMON/P/X1,X2,X3,SCL,SCS,SCI,I1,I2,I3,INV,LP,ISP,NRS
      DIMENSIONX1(500),SCL(500),X2(100),SCS(100),X3(100),SCI(100)
      COMPLEX UM(30,30),W(30),SST(30),CC,SS,CB,US(30,30)
      SS=0.
      CC=0.
      DO 71 M=1,NPL
      SS=SS+W(M)*CONJG(SST(M))
      CB=0.
      DO 70 N=1,NPL
 70   CB=CB+           W(N)*UM(M,N)
 71   CC=CC+CB*CONJG(W(M))
      SSR=CABS(SS)**2
      CCR=CABS(CC)
      SCO=SSR/CCR*CONST
      SCS(I2)=SCO
      IF(I.EQ.1) GO TO 5
      CC=0.
      DO 1 M=1,NPL
      CB=0.
      DO 2 N=1,NPL
 2    CB=CB+             W(N)*US(M,N)
 1    CC=CC+CONJG( W(M))*CB
      CCR=CABS(CC)
      SCL(I2)=SSR/CCR*CONST
 5    CONTINUE
      DO 6 M=1,NPL
      DO 6 N=1,NPL
 6    US(M,N)=UM(M,N)
      RETURN
      END


      SUBROUTINE QUANT1(V,NL,SAT,NPL,NPR,ITL)
C     QUANTIZATION SUBROUTINE FOR A/D CONVERTER OF FIGURE 3
      COMPLEX V(8,320)
      COMMON/NS/ICX,ICY,ICT
      DO 10 M = 1,NPL
      DO 10 N = 1,NPR
      ICT=ICT+1
      XX=REAL(V(M,N))
      XA=ABS(XX)
      YY=AIMAG(V(M,N))
      YA=ABS(YY)
      IF(XA.LT.SAT) GO TO 9
      XX=SAT*XX/XA
      ICX=ICX+1
    9 IF(YA.LT.SAT) GO TO 5
      YY=SAT*YY/YA
      ICY=ICY+1
 5    IF(ITL.EQ.0) GO TO 10
      ZX = (XX/2./SAT + .5)*(NL   )+.5
      ZY = (YY/2./SAT + .5)*(NL   )+.5
      XX = AINT(ZX)/(NL   )*2*SAT-SAT
      YY = AINT(ZY)/(NL   )*2*SAT-SAT
   10 V(M,N)=CMPLX(XX,YY)
      RETURN
      END


      SUBROUTINE QUANT2(V,NL,SAT,NPL,NPR,ITL)
C     QUANTIZATION SUBROUTINE FOR A/D CONVERTER OF FIGURE 3
      COMPLEX V(8,320)
      COMMON/NS/ICX,ICY,ICT
      DO 10 M = 1,NPL
      DO 10 N = 1,NPR
      AMP=CABS(V(M,N))
      ICT=ICT+1
      IF(AMP.LT.SAT) GO TO 5
      V(M,N)=V(M,N)*SAT/AMP
      ICX=ICX+1
 5    IF(ITL.EQ.0) GO TO 10
      XX=REAL(V(M,N))
      YY=AIMAG(V(M,N))
      ZX = (XX/2./SAT + .5)*(NL   )+.5
      ZY = (YY/2./SAT + .5)*(NL   )+.5
      XX = AINT(ZX)/(NL   )*2*SAT-SAT
      YY = AINT(ZY)/(NL   )*2*SAT-SAT
   10 V(M,N)=CMPLX(XX,YY)
      RETURN
      END


      SUBROUTINE MATINV(N,A)
      COMPLEX A(30,30)
      DO 11 N1=1,N
      DO 12 J=1,N
      IF(J.EQ.N1) GO TO 12
      A(N1,J)=A(N1,J)/A(N1,N1)
   12 CONTINUE
      DO 15 I=1,N
      IF(I.EQ.N1) GO TO 15
      DO 16 J=1,N
      IF(J.EQ.N1) GO TO 16
      A(I,J)=A(I,J)-A(I,N1)*A(N1,J)
   16 CONTINUE
      A(I,N1)=-A(I,N1)/A(N1,N1)
   15 CONTINUE
      A(N1,N1)=1./A(N1,N1)
   11 CONTINUE
      RETURN
      END
```

DISTRIBUTION LIST

Commander
Naval Air Systems Command
Department of the Navy
Washington, D. C. 20361
Attn: AIR-533D3 (1 copy quarterly) (3 copies final)
AIR-50174 (14 copies final only)
AIR-310B (2 copies quarterly) (5 copies final)

Commander
Naval Sea System Command
Department of the Navy
Washington, D. C. 20360
Attn: SEA-034

Commander
Department of the Air Force
Air Force Avionics Laboratory
Electronic Warfare Division
Wright-Patterson Air Force Base
Ohio 45433
Attn: Mr. Harold Weber

Commanding Officer
Naval Air Development Center
Warminster, Pennsylvania 18974
Attn: Mr. Jerry Guarini, AER-2

Director
Naval Research Laboratory
Washington, D. C. 20390
Attn: Mr. Fred Staudaher, Code 5368 (1 copy)
Mr. D. Stilwell, Code 7945 (1 copy)

Commander
Naval Weapons Center
China Lake, California 93555
Attn: Code 35
Code 40
Code 601

Commanding Officer
Naval Avionics Facility
Indianapolis, Indiana 46218
Attn: Mr. Paul Brink

Commander
Naval Electronics Laboratory Center
San Diego, California 92152
Attn: Code 2330 (2 copies)

General Dynamics/Electronics Division
P. O. Box 81127
San Diego, California 92138
Attn: Dr. G. Tricoles

Director
Electro-Sciences Laboratory
Ohio State University
1320 Kinnear Road
Columbus, Ohio 43212
Attn: Mr. Robert Fouty

Commander
Air Force Cambridge Research Laboratory
Laurence C. Hanscom Field
Bedford, Massachusetts 01730
Attn: Mr. Philip Blacksmith

Harry Diamond Laboratories
Microwave Research and Development Branch
Connecticut Avenue and Van Ness Street, N. W.
Washington, D. C. 20438
Attn: Mr. Howard S. Jones, Jr.

Syracuse University Research Corporation
Merrill Lane
University Heights
Syracuse, New York 13210
Attn: Mr. Sidney Applebaum

Dr. Norbert N. Bojarski
16 Circle Drive
Moorestown, New Jersey 08057

Hughes Aircraft Company
Radar Microwave Laboratory
Centinella and Teale
Culver City, California 90230
Attn: Dr. W. Kummer

Polytechnic Institute of New York
Department of Electrical Engineering and Electrophysics
Farmingdale, New York 11735
Attn: Dr. Alexander Hessel

Tennessee Technological University
College of Engineering
Cookeville, Tennessee 38501
Attn: Dr. Roy Adams

R. C. Hansen, Inc.
Suite 218
17100 Ventura Blvd.
Encino, California 91316

Naval Ammunition Depot
Crane, Indiana 47448
Attn: Mr. Rod C. Davis, Code 3083